

AD-A092 303

NAVAL POSTGRADUATE SCHOOL, MONTEREY CA
AN APPLICATION OF KALMAN FILTERING TO TORPEDO TRACKING.(U)
SEP 80 P A O'BRIEN

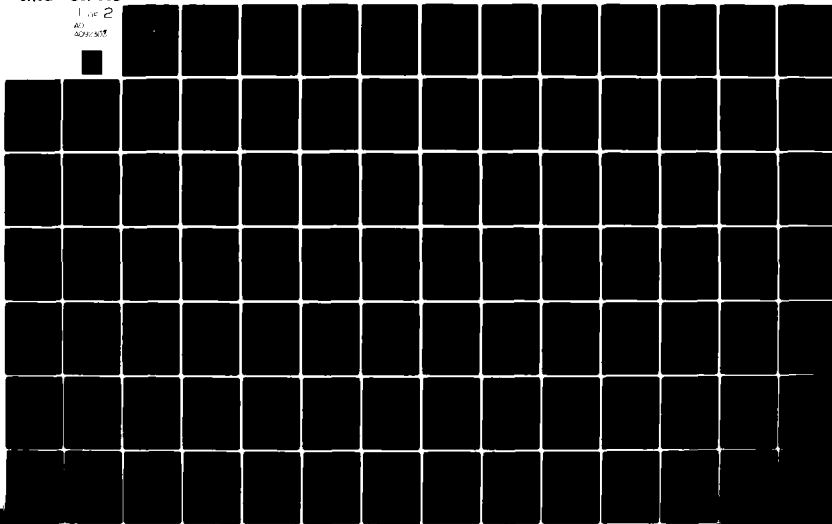
F/G 17/1

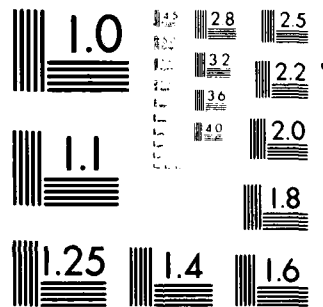
UNCLASSIFIED

NL

1 of 2

AD
A092303





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL II

②

NAVAL POSTGRADUATE SCHOOL
Monterey, California

R. S.

AD A092303



DTIC
ELECTE
S DEC 02 1980 **D**
E

THESIS

AN APPLICATION OF KALMAN FILTERING
TO TORPEDO TRACKING

by

Paul A. O'Brien

September 1980

Thesis Advisor:

H. A. Titus

Approved for public release; distribution unlimited

DDC FILE COPY

80 12 01 101

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A092 303	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN APPLICATION OF KALMAN FILTERING TO TORPEDO TRACKING.		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1980
7. AUTHOR(s) Paul A. O'Brien		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		12. REPORT DATE September 1980
		13. NUMBER OF PAGES 127
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Kalman Filtering Torpedo Tracking		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A sequential Extended Kalman filter routine was developed to provide real time estimates of torpedo position and depth on the three dimensional underwater tracking range at the Naval Torpedo Station, Keyport, Washington. Filter measurements consisted of acoustic pulse transit times from the torpedo to a receiving hydrophone array. These measurements, which are nonlinear functions of the position and depth coordinates, were linearized and		

20. (continued)

and the filter gains calculated on-line. Tests were conducted using simulated torpedo trajectories that traversed both single and multiple hydrophone arrays. It was found that filter performance was dependent on system noise and the distance the torpedo was from the hydrophone array. Position and depth errors ranged between 0 and 10 feet.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

Approved for public release; distribution unlimited

An Application of Kalman Filtering
To Torpedo Tracking

by

Paul A. O'Brien
B.S., University of Maryland, 1972

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1980

Author

Paul A. O'Brien

Approved by:

J. A. Titus Thesis Advisor
Alex Gerba Jr. Second Reader

R. K. Kish
Chairman, Department of Electrical Engineering

William M. Tolles
Dean of Science and Engineering

ABSTRACT

↓
A sequential Extended Kalman filter routine was developed to provide real time estimates of torpedo position and depth on the three dimensional underwater tracking range at the Naval Torpedo Station, Keyport, Washington. Filter measurements consisted of acoustic pulse transit times from the torpedo to a receiving hydrophone array. These measurements, which are nonlinear functions of the position and depth coordinates, were linearized and the filter gains calculated on-line. Tests were conducted using simulated torpedo trajectories that traversed both single and multiple hydrophone arrays. It was found that filter performance was dependent on system noise and the distance the torpedo was from the hydrophone array. Position and depth errors ranged between 0 and 10 feet.
↗

TABLE OF CONTENTS

	Page
I. INTRODUCTION	7
II. DESCRIPTION OF RANGE TRACKING GEOMETRY	9
III. THEORY--EXTENDED KALMAN FILTER	11
IV. PROBLEM DEFINITION--TORPEDO TRACKING WITH THE EXTENDED KALMAN FILTER	15
A. FILTER EQUATIONS	15
B. THE SEQUENTIAL EXTENDED KALMAN FILTER	23
V. TESTING AND SIMULATION	28
A. DESCRIPTION	28
B. EDITING ERRONEOUS TRANSIT TIME MEASUREMENTS	29
C. BOUNDING RESIDUAL BIAS ERRORS	30
D. MULTIPLE ARRAY TRACKING	32
VI. TEST RESULTS AND DISCUSSION	36
A. SERIES ONE--STRAIGHT RUNS	36
B. SERIES TWO--MANEUVERING RUNS	39
C. SERIES THREE--MULTIPLE ARRAY TRACKING	44
VII. CONCLUSIONS	45
FIGURES	47
APPENDIX A: PROGRAM DESCRIPTION AND FIGURES	98
A. PROGRAM SUBROUTINES	100
B. UTILITY PROGRAMS	102
APPENDIX B: SEQUENTIAL EXTENDED KALMAN FILTER PROGRAM LISTING	103
LIST OF REFERENCES	126
INITIAL DISTRIBUTION LIST	127

ACKNOWLEDGEMENT

The author is deeply indebted to Professor Hal Titus for his counsel and professional guidance during this project.

I. INTRODUCTION

The Naval Torpedo Station (NAVTORPSTA) Keyport, Washington currently operates two three-dimensional (3-D) underwater tracking ranges with acoustical capability. The underwater tracking range utilizes a sonar transmitter installed in the torpedo to be tracked. The transmitter is synchronized with a master clock. Timed acoustic pulses are received by bottom mounted hydrophone arrays and then relayed via cable to a computer at the observation site. The computer calculates the positional coordinates of the torpedo and plots its trajectory through the water.

The measured data, which consists of the elapsed time from transmission of a pulse until its receipt at the hydrophone array, is corrupted with noise due to the combined effects of environmental factors and measurement instruments.

In order to achieve a more accurate plot of the torpedo path, editing and smoothing techniques have been applied to the data during post run analyses.

To improve the real-time capability of the 3-D tracking ranges, Kalman filtering theory has been applied to the computer generated X, Y, and Z positional coordinates but is not operational [1].

In the near future an updated computer system composed of three MODCOMPIV computers will be in operation. The

majority of the software development requires interfacing current tracking programs and other related programs to the new computer system.

An opportunity exists for expanding the capability of the tracking system by applying a Kalman filter routine which processes measurements of the transit times of the acoustic pulses and provides an estimate of the position of the tracked object in real time. Prior research in this area [2], revealed that a Kalman filter utilizing a sequential estimation approach was desirable.

The intention is to develop and test a sequential Kalman filter tracking algorithm that can be interfaced with the current underwater range system.

II. DESCRIPTION OF RANGE TRACKING GEOMETRY

The hydrophone array, consisting of four independent elements, defines an orthogonal coordinate system in which transit time measurements are made. As shown in Figure 1, four hydrophones X, Y, Z, and C are on four adjacent vertices separated by a distance d , along the edge of the cube. The origin of the array coordinates is at the center of the cube with the orthogonal coordinates parallel to its edge. Positional information is computed from the transit times of a periodic synchronous acoustic signal travelling from the torpedo to the four hydrophones on the array. The range measures the tracked torpedo's position every 1.31 seconds to an accuracy that is typically within 3 to 30 feet. A more detailed description of the range tracking capability is described in [3].

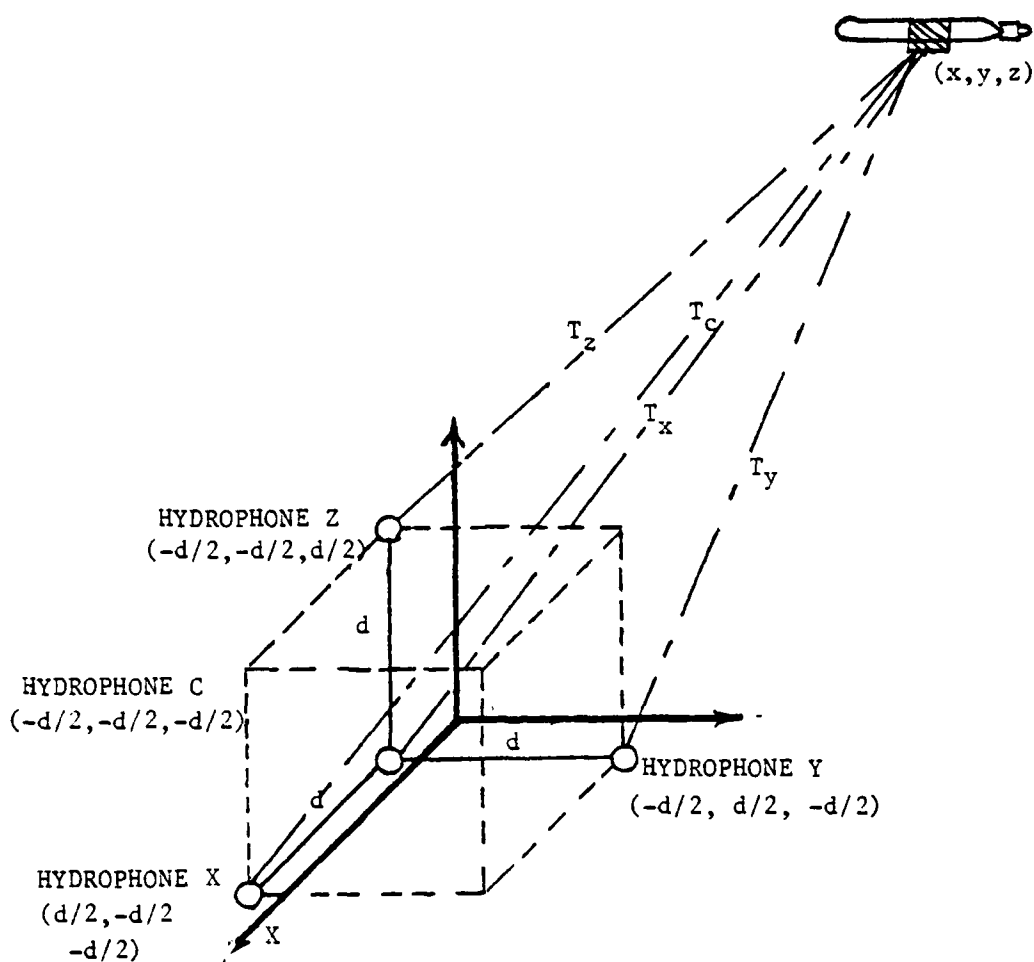


Figure 1. Geometry of a Tracking Array

III. THEORY--EXTENDED KALMAN FILTER

The equations describing the nonlinear relationship between transit times T_C , T_X , T_Y , T_Z and the position of the tracked torpedo are given by [3]

$$T_C = \frac{1}{VEL} [(x + \frac{d}{2})^2 + (y + \frac{d}{2})^2 + (z + \frac{d}{2})^2]^{\frac{1}{2}}, \quad (3.1)$$

$$T_X = \frac{1}{VEL} [(x - \frac{d}{2})^2 + (y + \frac{d}{2})^2 + (z + \frac{d}{2})^2]^{\frac{1}{2}}, \quad (3.2)$$

$$T_Y = \frac{1}{VEL} [(x + \frac{d}{2})^2 + (y - \frac{d}{2})^2 + (z + \frac{d}{2})^2]^{\frac{1}{2}}, \quad (3.3)$$

$$T_Z = \frac{1}{VEL} [(x + \frac{d}{2})^2 + (y + \frac{d}{2})^2 + (z - \frac{d}{2})^2]^{\frac{1}{2}}, \quad (3.4)$$

where $d = 30$ feet and $VEL = 4860$, which is approximately the average velocity of propagation of sound in water at Dabob Bay.* Since the transit times were readily available and are nonlinear functions of position, these equations can be linearized and Kalman Filter theory applied using the Extended Kalman Filter [2]. This procedure produces a real time measurement, with filtering on the corrupted transit times T_C , T_X , T_Y , and T_Z without the necessity of converting these times to positions.

*Sound velocity in water varies as a function of temperature, salinity, and depth.

For the three-dimensional location problem three position states (x, y, z) and two velocity states (v_x, v_y) specify target motion. The discrete linear and nonlinear observation equations are given by

$$\underline{x}(k+1) = \Phi \cdot \underline{x}(k) + \Gamma \cdot \underline{w}(k) \quad (3.5)$$

and

$$z(k) = \underline{h}(\underline{x}(k), k) + \underline{v}(k) \quad (3.6)$$

In these equations Φ and Γ are constant matrices and \underline{h} is a nonlinear function of the state variable \underline{x} . $\underline{w}(k)$ is plant excitation noise and $\underline{v}(k)$ is measurement noise. The plant noise and measurement noise are assumed uncorrelated (white) with zero mean.

That is,

$$E[\underline{w}(k) \cdot \underline{w}^T(j)] = Q'(k) \delta_{kj}$$

and

$$E[\underline{v}(k) \cdot \underline{v}^T(j)] = R(k) \delta_{kj}$$

with

$$\begin{aligned} \delta_{kj} &= 1, \quad k = j \\ &= 0, \quad k \neq j \end{aligned}$$

In order to apply the linear filter equation (3.6) is expanded in a Taylor series about the best estimate of the

the state at that time and only the first-order terms are kept.

Equation (3.6) gives

$$z(k) = H(k) \cdot \underline{x}(k) + \underline{v}(k) \quad (3.7)$$

where

$$H(k) = \left. \frac{\partial h}{\partial \underline{x}} \right|_{\underline{x}'(k) = \hat{\underline{x}}(k/k-1)} \quad (3.7a)$$

$\hat{\underline{x}}(k/k-1)$ is the predicted value of the state before the k^{th} measurement.

A state error vector is defined by

$$\tilde{\underline{x}} = \hat{\underline{x}}(k) - \underline{x}(k),$$

and a predicted state error vector is defined by

$$\tilde{\underline{x}}(k/k-1) = \hat{\underline{x}}(k/k-1) - \underline{x}(k).$$

The covariance of state error matrix is defined by

$$P(k) = E[\tilde{\underline{x}}(k) \cdot \tilde{\underline{x}}^T(k)],$$

and the predicted covariance of state error matrix is given by

$$P(k/k-1) = E[\tilde{\underline{x}}(k/k-1) \cdot \tilde{\underline{x}}^T(k/k-1)].$$

The state excitation matrix is given by

$$Q(k) = \Gamma(k) E[\underline{w}(k) \cdot \underline{w}^T(k)] \cdot \Gamma^T(k)$$

and the measurement noise covariance matrix is

$$R(k) = E[\underline{v}(k) \cdot \underline{v}^T(k)]$$

The Kalman filter equations are given by [4,5]:

$$P(k+1/k) = \Phi P(k/k) \Phi^T + Q(k) \quad (3.8a)$$

$$G(k) = P(k/k-1)H^T(k) [H(k) \cdot P(k/k-1)H^T(k) + R(k)]^{-1} \quad (3.8b)$$

$$P(k) = [I - G(k)H(k)]P(k/k-1) \quad (3.8c)$$

$$\hat{\underline{x}}(k+1/k) = \Phi \hat{\underline{x}}(k/k) \quad (3.8d)$$

$$z(k/k-1) = h(\hat{\underline{x}}(k/k-1), k) \quad (3.8e)$$

$$\hat{\underline{x}}(k) = \hat{\underline{x}}(k/k-1) + G(k) [z(k) - \underline{z}(k/k-1)] \quad (3.8f)$$

The Q matrix serves not only to allow for maneuvering but also to account for any model inaccuracies, that is, any discrepancies between the true action of the torpedo and its characterization by Equation (3.5). The Q also serves to prevent the gain matrix G(k) from approaching zero by always insuring uncertainty in the predicted covariance of error matrix P(k+1/k).

IV. PROBLEM DEFINITION--TORPEDO TRACKING WITH THE EXTENDED KALMAN FILTER

A. FILTER EQUATIONS

Figure 2 illustrates the geometry of the states used for the filter. For a constant course, constant-speed torpedo, the plant state equations can be described as two second order systems, one for X and one for Y. The depth (Z) of the torpedo is maintained constant and any velocity in the Z direction is assumed to be random, uncorrelated with zero mean. Z is therefore described as a first order system.

The plant state equations are

$$\begin{bmatrix} x(k+1) \\ v_x(k+1) \\ y(k+1) \\ v_y(k+1) \\ z(k+1) \end{bmatrix} = \begin{bmatrix} x(k) + Tv_x(k) + g_1(\gamma_{\dot{\theta}_t}, \gamma_{\dot{v}_t}, k) \\ v_x(k) + g_2(\gamma_{\dot{\theta}_t}, \gamma_{\dot{v}_t}, k) \\ y(k) + Tv_y(k) + g_3(\gamma_{\dot{\theta}_t}, \gamma_{\dot{v}_t}, k) \\ v_y(k) + g_4(\gamma_{\dot{\theta}_t}, \gamma_{\dot{v}_t}, k) \\ z(k) + g_5(\gamma_{\dot{z}}) \end{bmatrix} \quad (4.1)$$

where $x(k)$, and $y(k)$ are the position coordinates at time $t(k)$, $v_x(k)$ and $v_y(k)$ are the X and Y components of velocity. T is the time between observations, i.e. $t(k+1)-t(k)$ and $z(k)$ is the depth of the torpedo.

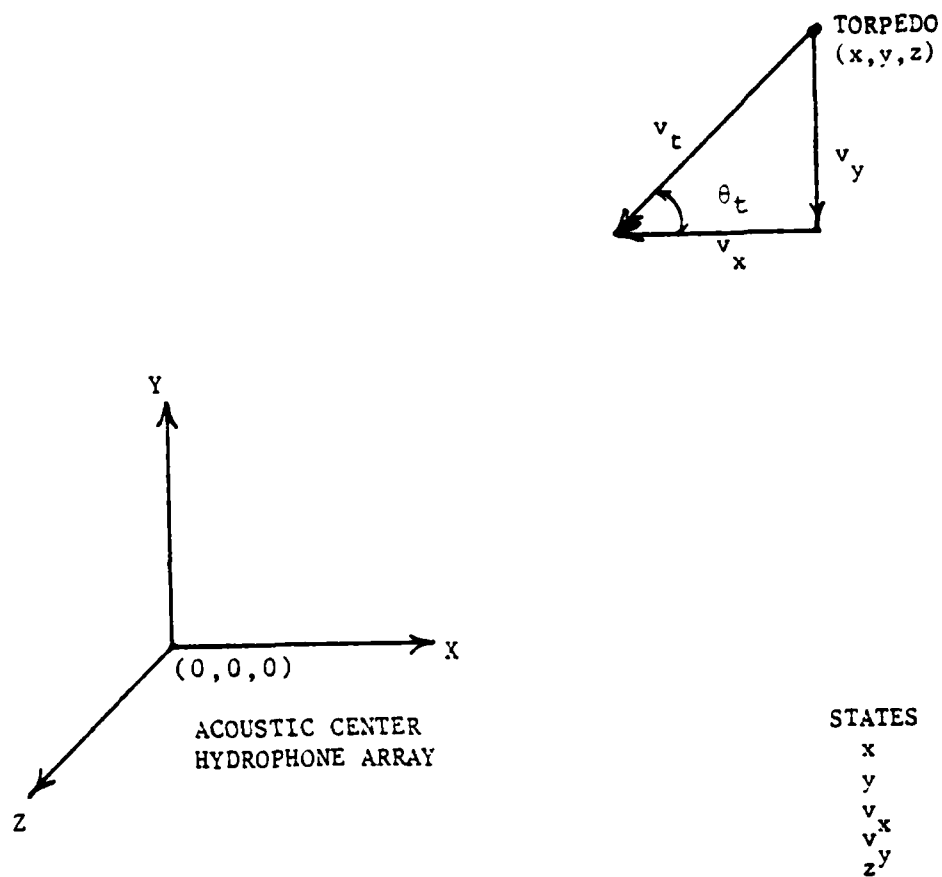


Figure 2. Filter Geometry

The excitation terms g_1 through g_5 are included to account for the random changes in speed, heading, and depth which can occur in a maneuvering torpedo. The quantities γ_{v_t} , γ_{θ_t} , and γ_z are the random changes of the torpedo which are assumed to be independent, zero mean, rates of change. They have variances defined by

$$\sigma_{\dot{v}_t}^2 = E[\gamma_{v_t}^2]$$

$$\sigma_{\dot{\theta}_t}^2 = E[\gamma_{\theta_t}^2]$$

$$\sigma_{\dot{z}}^2 = E[\gamma_z^2]$$

The values of the standard deviations are taken from typical maneuvering parameters for the torpedo;

$$\sigma_{\dot{\theta}_t} = 22^\circ/\text{sec}$$

$$\sigma_{\dot{v}_t} = 36 \text{ ft/sec}^2$$

$$\sigma_{\dot{z}} = 1 \text{ ft/sec}$$

The effect of this excitation is to increase the predicted value of the covariance of the state error matrix.

Writing the equations in state form results in

$$x(k+1) = \Phi x(k) + \Gamma w(k) \quad (4.2)$$

where

$$\Phi = \begin{bmatrix} 1 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\Gamma = \begin{bmatrix} T^2/2 & 0 & 0 \\ T & 0 & 0 \\ 0 & T^2/2 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix}$$

The vector $\underline{w}(k)$ represents the effect on the states of the random excitations. Reference 5 has shown that the vector may be calculated from the equations relating torpedo X and Y velocity to the torpedo heading and velocity. The X velocity is

$$\dot{x} = v_x = v_t \cos \theta_t$$

which when differentiated gives

$$\ddot{x} = -v_t \dot{\theta}_t \sin \theta_t + \dot{v}_t \cos \theta_t \quad (4.4)$$

where

$$\dot{\theta}_t = \gamma \dot{\theta}_t$$

$$\dot{v}_t = \gamma \dot{v}_t$$

$$\sin \theta_t = \frac{v_y}{v_t}$$

$$\cos \theta_t = \frac{v_x}{v_t}$$

and upon substitution of the above relationship in equation (4.4),

$$\ddot{x} = -v_y \gamma \dot{\theta}_t + \gamma \dot{v}_t \frac{v_x}{v_t} \quad (4.5)$$

Similarly

$$\dot{y} = v_y = v_t \sin \theta_t \quad (4.6)$$

$$\ddot{y} = v_t \dot{\theta}_t \cos \theta_t + \dot{v}_t \sin \theta_t$$

and after substitution

$$\ddot{y} = v_x \gamma \dot{\theta}_t + \frac{v_y}{v_t} \gamma \dot{v}_t \quad (4.8)$$

The depth term is just

$$z = \gamma \dot{z} \quad (4.9)$$

Thus

$$\underline{w}(k)^T = [(\gamma_{\dot{v}_t} \frac{v_x}{v_t} - \gamma_{\dot{\theta}_t} v_{\dot{y}}) (\gamma_{\dot{\theta}_t} v_x + \gamma_{\dot{v}_t} \frac{v_y}{v_t}) \gamma_{\dot{z}}] \quad (4.10)$$

where from the assumption on the γ 's

$$E[\underline{w}(k)] = 0 \quad (4.11)$$

The excitation covariance matrix is thus found from

$$Q = EE[\underline{w}(k)\underline{w}(k)^T] \Gamma^T \quad (4.12)$$

Let

$$\sigma_{\ddot{x}}^2 = (\frac{v_x}{v_t})^2 \sigma_{\dot{v}_t}^2 + v_y^2 \sigma_{\dot{\theta}_t}^2 \quad (4.13a)$$

$$\sigma_{\ddot{y}}^2 = (\frac{v_y}{v_t})^2 \sigma_{\dot{v}_t}^2 + v_x^2 \sigma_{\dot{\theta}_t}^2 \quad (4.13b)$$

$$\sigma_{\ddot{x}\ddot{y}} = v_x v_y [\frac{\sigma_{\dot{v}_t}^2}{v_t^2} - \sigma_{\dot{\theta}_t}^2] \quad (4.13c)$$

where the states are evaluated at the current state estimates $\hat{x}(k)$. Substituting these expressions in the Q matrix results in

$$Q = \begin{bmatrix} (T^2/2)\sigma_{\ddot{x}}^2 & T^3/2\sigma_{\ddot{x}}^2 & (T^2/2)\sigma_{\ddot{x}\ddot{y}}^2 & T^3/2\sigma_{\ddot{x}\ddot{y}}^2 & 0 \\ & T^2\sigma_{\ddot{x}}^2 & T^3/2\sigma_{\ddot{x}\ddot{y}}^2 & T^2\sigma_{\ddot{x}\ddot{y}}^2 & 0 \\ & & (T^2/2)\sigma_{\ddot{y}}^2 & T^3/2\sigma_{\ddot{y}}^2 & 0 \\ & & & T^2\sigma_{\ddot{y}}^2 & 0 \\ \text{Symmetric} & & & & T^2\sigma_{\ddot{z}}^2 \end{bmatrix} \quad (4.14)$$

The excitation matrix serves not only to take into account the possibility of maneuvering, but of model inaccuracies as well [5]. Q is also used to prevent the gains of the filter from approaching zero as more and more data is processed, by insuring some uncertainty in the predicted state values.

The observation equations are nonlinear in the states and are given by

$$z(k) = \begin{bmatrix} T_C(k) \\ T_X(k) \\ T_Y(k) \\ T_Z(k) \end{bmatrix} = \begin{bmatrix} \frac{1}{VEL} [(x(k)+d/2)^2 + (y(k)+d/2)^2 + (z(k)+d/2)^2]^{\frac{1}{2}} + v(k) \\ \frac{1}{VEL} [(x(k)-d/2)^2 + (y(k)+d/2)^2 + (z(k)+d/2)^2]^{\frac{1}{2}} + v(k) \\ \frac{1}{VEL} [(x(k)+d/2)^2 + (y(k)-d/2)^2 + (z(k)+d/2)^2]^{\frac{1}{2}} + v(k) \\ \frac{1}{VEL} [(x(k)+d/2)^2 + (y(k)+d/2)^2 + (z(k)-d/2)^2]^{\frac{1}{2}} + v(k) \end{bmatrix} \quad (4.15)$$

The measurement noises, $v(k)$'s, are assumed to be zero-mean and independent with a covariance matrix

$$R(k) = \begin{bmatrix} \sigma_{T_C}^2 & 0 & 0 & 0 \\ 0 & \sigma_{T_X}^2 & 0 & 0 \\ 0 & 0 & \sigma_{T_Y}^2 & 0 \\ 0 & 0 & 0 & \sigma_{T_Z}^2 \end{bmatrix} \quad (4.16)$$

The magnitude of the transit times noise is a function of the signal-to-noise ratio at the hydrophone array, which is influenced by several environmental factors.

Discussions with NAVTORPSTA personnel indicate a typical accuracy of ± 10 microseconds and this value is used as the standard deviation for the simulation runs.

Equation (3.7a) can be used to give the linearized observation matrix. The result is

$$H(k) = \begin{bmatrix} \frac{\partial T_C}{\partial x} & \frac{\partial T_C(k)}{\partial v_x} & \frac{\partial T_C(k)}{\partial y} & \frac{\partial T_C(k)}{\partial v_y} & \frac{\partial T_C(k)}{\partial z} \\ \frac{\partial T_X(k)}{\partial x} & \frac{\partial T_X(k)}{\partial v_x} & \frac{\partial T_X(k)}{\partial y} & \frac{\partial T_X(k)}{\partial v_y} & \frac{\partial T_X(k)}{\partial z} \\ \frac{\partial T_Y(k)}{\partial x} & \frac{\partial T_Y(k)}{\partial v_x} & \frac{\partial T_Y(k)}{\partial y} & \frac{\partial T_Y(k)}{\partial v_y} & \frac{\partial T_Y(k)}{\partial z} \\ \frac{\partial T_Z(k)}{\partial x} & \frac{\partial T_Z(k)}{\partial v_x} & \frac{\partial T_Z(k)}{\partial y} & \frac{\partial T_Z(k)}{\partial v_y} & \frac{\partial T_Z(k)}{\partial z} \end{bmatrix} \quad (4.17)$$

When the derivatives are taken and evaluated at the predicted state values $x(k/k-1) = x'(k)$ the result is

$$H(k) = \begin{bmatrix} \frac{x'(k)+d/2}{DEN1} & 0 & \frac{y'(k)+d/2}{DEN1} & 0 & \frac{z'(k)+d/2}{DEN1} \\ \frac{x'(k)-d/2}{DEN2} & 0 & \frac{y'(k)+d/2}{DEN2} & 0 & \frac{z'(k)+d/2}{DEN2} \\ \frac{x'(k)+d/2}{DEN3} & 0 & \frac{y'(k)-d/2}{DEN3} & 0 & \frac{z'(k)+d/2}{DEN3} \\ \frac{x'(k)+d/2}{DEN4} & 0 & \frac{y'(k)+d/2}{DEN4} & 0 & \frac{z'(k)-d/2}{DEN4} \end{bmatrix} \quad (4.18)$$

where

$$DEN1 = [(x'(k)+d/2)^2 + (y'(k)+d/2)^2 + (z'(k)+d/2)^2]^{\frac{1}{2}}$$

$$DEN2 = [(x'(k)-d/2)^2 + (y'(k)+d/2)^2 + (z'(k)+d/2)^2]^{\frac{1}{2}}$$

$$DEN3 = [(x'(k)+d/2)^2 + (y'(k)-d/2)^2 + (z'(k)+d/2)^2]^{\frac{1}{2}}$$

$$DEN4 = [(x'(k)+d/2)^2 + (y'(k)+d/2)^2 + (z'(k)-d/2)^2]^{\frac{1}{2}} .$$

The ϕ matrix, Q matrix, R matrix, and H matrix are then used in the Kalman filter equations (3.8).

B. THE SEQUENTIAL EXTENDED KALMAN FILTER

The Kalman filter equations described in the previous section represent a system that is characterized by a one-step estimation and prediction for each set of four transit time observables (T_C , T_X , T_Y , and T_Z). This one step process exhibits the following salient features:

1. The four transit times received during each 1.31 second measurement interval (time slot) are processed simultaneously by the filter equations. Updated covariance of error and state estimates are therefore obtained only once for each time slot.

2. A major part of the time period for the estimation calculation is spent in the gain equation where the inversion of a 4×4 matrix is required.

3. The Extended Kalman filter is based on a linearization about the predicted value of a period t_k to t_{k+1} . Unacceptably large or erroneous transit time measurements will cause the filter to ignore the update information for the entire time slot. The predicted estimate will become the new estimate for the time slot and the interval extended.

If one assumes that each of the four hydrophones in the array are statistically independent of one another, a sequential approach can be taken to process each of the four transit times separately. Thus, each observable will be processed in sequence, and the result of processing one measurement is used in the following computation to process the next measurement until all four observables have been used.

The advantages of the sequential approach over the traditional one-step estimation and prediction are:

1. Since the four transit times are independent and processed sequentially, the covariance of error matrix and

the state vector are updated four times during each time slot. Thus more accurate estimates of the filter states are achieved.

2. Modification of the filter equations for the sequential approach circumvented the matrix inversion in the gain equation. Therefore filter cycle time is reduced substantially.

3. An invalid transit time measurement will result in the filter ignoring the update information for that particular measurement only. For example, if only three of the four transit times are acceptable, the filter would be sequentially updated three times during that particular time slot. Thus, the filter is less sensitive to erroneous transit time measurements.

To utilize the sequential Extended Kalman filter, equations (3.8) must be modified. Calculations are performed on each of the four independent transit times in the following order: T_C , T_X , T_Y , and T_Z for each 1.31 second time slot. The estimate of the states $x(k/k)$, based on one transit time measurement are used as the prediction $x(k/k-1)$ for the calculations on the next measurement. Thus for the first time measurement T_C only the first row of the linearizing H matrix is calculated.

Next the first gain column corresponding to the first time measurement T_C is calculated by

$$G_{icol} = \frac{P(k/k-1)H_{irow}^T}{H_{irow}P(k/k-1)H_{irow}^T + R_{ii}} \quad (4.19)$$

where $i = 1$ to 4 corresponding to the four measured transit times. Thus, the first row of the H matrix is used to calculate the first column of the gain matrix with both corresponding to the first measured time T_C .

Next, an estimate of the particular observation time is calculated using equation (3.8e) evaluated at the predicted state $\underline{x}(k/k-1)$.

The difference between observed transit time and the estimated transit times forms the residual which is used in the estimate equation

$$\underline{x}_i = \underline{x}(k/k-1) + G_{icol}[\text{Residual}] \quad (4.20)$$

This equation gives an estimate of the states based on one measurement.

Next, the covariance of error is calculated based on one measurement by

$$P_i = [I - G_{icol}H_{irow}]P_{i-1} \quad (4.21)$$

where

I = identity matrix

P_{i-1} = the covariance matrix calculated from the previous transit time measurement or if $i = 1$, the prediction $P(k/k-1)$.

After the first iteration, \bar{x}_1 becomes $\underline{x}(k/k-1)$ and P_1 becomes $P(k/k-1)$ for the second iteration which calculates the estimate of the states based on the second measurement T_X .

After four iterations ($k = 4$), \underline{x}_4 becomes the estimate for the time slot $x(k/k)$ and P_4 becomes the covariance of error $P(k/k)$.

The predictions for the next time slot are calculated using equations (3.8a) and (3.8d). This process is repeated for each time slot.

IV. TESTING AND SIMULATION

A. DESCRIPTION

The sequential Extended Kalman filter routine is tested using simulated torpedo tracks generated by the IBM-360 computer. A variety of track scenarios are produced to test the filter performance during single array tracking.

The first series of tests used constant course and speed tracks which transitioned multiple array quadrants including:

1. crossing north of the array
2. crossing above the array
3. crossing diagonally through the array

The next series of tests adds a number of typical torpedo maneuvers to the constant course and speed tracks. These maneuvers consisted of 10 and 20 °/sec turns with G forces ranging from 1/4-G to 1-G.

For the above tests filter initialization errors range from 0 to 40 feet/sec for X and Y velocity, and 0 to 60 feet for X and Y position. The Z depth is maintained constant at 300 feet which is approximately the average depth of water at Dabob bay. Zero mean Gaussian noise is added to corrupt the observed transit times for all runs.

The last series of tests demonstrates the ability of the filter to track through the areas of multiple arrays.

Before the above series of tests can be conducted, it is necessary to add operational features to the software which

provide the following capability:

1. detect erroneous transit time measurements
2. minimize position error during high dynamic maneuvers
3. track through multiple arrays

A more detailed explanation of these features is described below.

B. EDITING ERRONEOUS TIME MEASUREMENTS

The operation of the filter may be adversely affected by erroneous transit time measurements. Large errors in the transit time measurements will produce errors in the state vector estimate and the linearizing H matrix. This may cause the filter to become unstable and exhibit divergent characteristics. In some extreme cases catastrophic failure may result. Measurement errors can occur because of many factors including an error in the transit time of the acoustic pulse primarily due to the receipt of multi path signals from previous time slots that have reflected off the surface, bottom, or different density layers in the water.

To guard against catastrophic failures a three-sigma gate using the covariance of measurement noise (R) and the covariance of estimation error ($P(k/k)$) is implemented. For each calculation of a state estimate ($\underline{x}(k/k)$), the largest positional covariance of error is used, either \underline{x} , \underline{y} , or \underline{z} , and converted to time in seconds using the average velocity of sound in water for Dabob bay, 4860 ft/sec. The gate then is

written for each time measurement $i = 1$ to 4:

$$\text{Gate} = \frac{\sqrt{(P_{jj})_{\text{maximum}} + R_{ii}}}{4860}$$

where $j = 1, 3, 5$. The gate expands or decreases depending on the confidence level of the position estimate and the transit time. If ZDIFF which is the difference between the actual transit time received and the predicted transit time to a particular hydrophone exceeds the gate, the measurement is considered unacceptable and the filter gain is set to zero causing the filter to ignore the data and take the prediction of the states as the estimate

$$\underline{x}(k/k) = \underline{x}(k/k-1)$$

An invalid time measurement zeros only the gain column for that particular hydrophone causing only that hydrophone's data to be ignored.

C. BOUNDING RESIDUAL BIAS ERROR

As previously mentioned, the torpedo dynamics are modeled in the filter's excitation covariance matrix Q .

During preliminary tests it becomes evident that torpedo turn rates between 5 and 20°/sec yield position errors ranging from approximately 10 to 40 ft. Increasing the standard deviation parameters (σ_{θ} , σ_{v_t} , σ_z) for the adaptive Q matrix helps to a certain extent, however as expected a point is reached where the noise starts to become the dominant error source. There are two major reasons why the filter develops errors during turns:

1. The filter's predicted state estimates are computed assuming a constant velocity over a 1.31 second sampling interval. During a turn the velocity vector of the torpedo is continuously rotating, therefore the filter's predicted estimate will be in error. A large magnitude error will require more than one time slot to remove. A faster sampling rate would reduce the error and may alleviate the problem; however, the Dabob tracking range system has been designed for a 1.31 second sampling rate and cannot be changed.

2. The adaptive Q matrix assumes that the torpedo accelerations are normally distributed and uncorrelated. Torpedo tracking maneuvers do not fit these assumptions.

To minimize position errors during maneuvers a procedure has been developed in order that repeated estimates of the state vector are obtained during a single sampling period (1.31 second time slot) until the transit time residual error falls below a preset threshold.

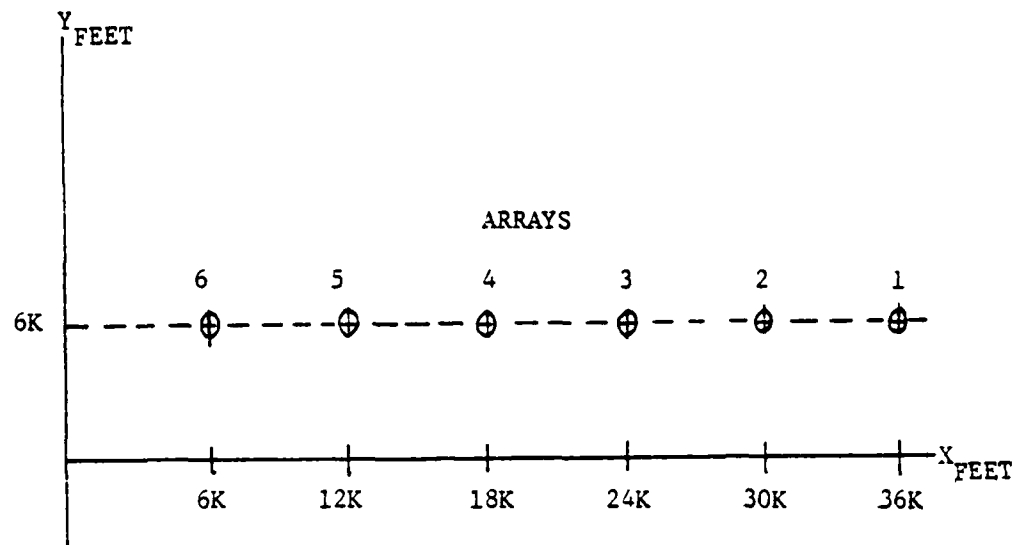
In Section IV it was stated that during each time slot the gain, covariance, and state estimate equations (4.19-4.21) were iteratively solved for each of the four sequential transit times T_C , T_X , T_Y , and T_Z . Subsequently additional software was incorporated to calculate the estimated transit time to each of the four hydrophones based on the last updated state vector. Four time residuals are then computed by subtracting the estimated time to each hydrophone from the observed transit time. The average of the absolute value of

the time residuals is then formed. If the average residual error exceeds the preset threshold in the software, which is indicative of large error in the state vector, Q is calculated and added to the last updated covariance of error matrix P . The filter then reiterates the gain, covariance, and state estimate equations for the same time slot. If the new state vector generates an average residual error still exceeding the threshold value, then the process is repeated. This procedure continues until the average residual error falls below the preset threshold at which time an acceptable state vector estimate has been obtained for the time slot. Once this occurs the prediction equations (3.8a) and (3.8d) are solved and the filter proceeds to the next time slot and commences processing new transit time measurements. As will be shown later the filter did successfully bound the position error during turning maneuvers. For these tests the threshold constant is set at 10 microseconds, which is the value of RMS measurement noise.

C. MULTIPLE ARRAY TRACKING

Initial tests were performed on tracks in the area of one array. In order to more closely simulate a typical run on the range, a scheme was designed to track the torpedo through multiple arrays [2].

First, a coordinate system is defined as shown in Figure 3. The center of the coordinate system is geographically



Coordinate System for Multiple Array Tracking

	C HYDRO			X HYDRO			V HYDRO			Z HYDRO		
	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
A R R A Y S	36000	6000	0	36030	6000	0	36000	6030	0	36000	6000	30
	30000	6000	0	30030	6000	0	30000	6030	0	30000	6000	30
	24000	6000	0	24030	6000	0	24000	6030	0	24000	6000	30
	18000	6000	0	18030	6000	0	18000	6030	0	18000	6000	30
	12000	6000	0	12030	6000	0	12000	6030	0	12000	6000	30
	6000	6000	0	6030	6000	0	6000	6030	0	6000	6000	30

HYDRO--Hydrophone Location Matrix

Figure 3

near the entrance to Dabob bay in the simulation. Array number 6 is the closest array to the coordinate center. In the simulation array 1 is at 36,000 feet from coordinate center and array 6 is 6,000 feet. The C hydrophone is assumed to be the axis location of each array. Then each X position for the X hydrophone in each array is $X_C + 30$, each Y position for the Y hydrophone is $Y_C + 30$ and each Z position for the Z hydrophone is $Z_C + 30$. These 72 positions, an XYZ position for each of 4 hydrophones in 6 arrays, are placed into a 6 x 12 matrix HYDRO and referenced throughout the routine. The torpedo position is referenced to a central level rectangular coordinate system. The nonlinear observation equations become

$$z(k) = \begin{bmatrix} T_C(k) \\ T_X(k) \\ T_Y(k) \\ T_Z(k) \end{bmatrix} = \begin{bmatrix} \frac{1}{VEL} [(x'(k) - X_{iC})^2 + (y'(k) - Y_{iC})^2 + (z'(k) - Z_{iC})^2]^{\frac{1}{2}} \\ \frac{1}{VEL} [(x'(k) - X_{iX})^2 + (y'(k) - Y_{iX})^2 + (z'(k) - Z_{iX})^2]^{\frac{1}{2}} \\ \frac{1}{VEL} [(x'(k) - X_{iY})^2 + (y'(k) - Y_{iY})^2 + (z'(k) - Z_{iY})^2]^{\frac{1}{2}} \\ \frac{1}{VEL} [(x'(k) - X_{iZ})^2 + (y'(k) - Y_{iZ})^2 + (z'(k) - Z_{iZ})^2]^{\frac{1}{2}} \end{bmatrix}$$

where $T_C(k)$, $T_X(k)$, $T_Y(k)$, and $T_Z(k)$ are the transit times to each of the four hydrophones and $X'(k)$, $Y'(k)$, and $Z'(k)$ are the filter estimates $x'(k/k)$ evaluated at the predicted positions $\underline{x}(k/k-1)$. The subscripted variables X, Y, and Z are the coordinates of a particular array being used.

The decision parameter used to determine the switching from array to array is a straight handoff. If the predicted x position is greater than 3,000 feet from the array in use, then an index (I8) is incremented and the next row of HYDRO is implemented. This placed into the routine the X, Y, and Z positions of the hydrophones in the next array. The handoff can easily be utilized in real range operations, as the transit times from adjacent arrays are present at the computer for a particular time slot. For simulation, it is assumed that in all the arrays each axis pointed in the same direction. In actual range operations each array is tilted about both the X and Y axis. Since the true transit times are derived in a tilted coordinate frame, the filter's estimate of transit time must also be calculated in a tilted coordinate frame. The tilt angle measurements along with the level rectangular coordinates of the array with respect to the central rectangular coordinate system can be input into the matrix HYDRO to rotate the coordinates of each hydrophone in the array.

VI. TEST RESULTS AND DISCUSSION

A. SERIES ONE--STRAIGHT RUNS

This series of tests includes constant speed and heading tracks. Initial position and velocity errors are set to zero. Measurement noise is added to all runs. Torpedo velocity ranged from 20 to 55 knots. Figure 4 shows the true trajectory of the torpedo in the horizontal X-Y plane. The center of the hydrophone array is located at positional coordinates (0,0). Figure 5 describes the filter's RMS estimate of position error as a function of sampling time. The RMS estimates are obtained by taking the square-root of the appropriate diagonal terms of the covariance matrix. It should be noted that the propagation of RMS estimates are a function of the position of the torpedo with respect to the hydrophone array. As shown in Figure 4, if $Y = 0$, the torpedo path is inbound toward the array along the "X" axis. Upon examining Figure 5, it is evident that the RMS estimate of X position error or the uncertainty in the filter's X position estimate is small and relatively constant throughout the entire trajectory. The uncertainty in the filter's Y position estimate decreases as the torpedo approaches the array, becomes a minimum at the closest point of approach (CPA), and increases as the torpedo moves outbound away from the array. This behavior in the filter's RMS estimates can be explained by recognizing

that the cluster of four hydrophones in the array actually describe a position fixing system. The torpedo position is located by the intersection of circular lines of position (range arcs) originating from each hydrophone on the array. In general, the greater distance the torpedo is located from the array, the shallower the range arc intersections become and the more uncertainty exists in the actual torpedo position. For the case where the torpedo is moving along the X axis, the uncertainty in position exists only in the Y direction. If the torpedo's X distance from the array becomes infinite, the range arcs from each hydrophone appear to originate from a single point source. The position fixing system then reduces to a single dimension ranging system along the X axis with the uncertainty in the Y position estimate becoming infinite. If the torpedo approaches the array along the Y axis ($X = 0$), the filter's RMS estimate of Y position error will be relatively small, and as the torpedo's Y distance from the array becomes infinite, the filter's uncertainty in the X position estimate becomes infinite.

In order to keep RMS position error estimates within allowable tolerance the Dabob range tracking system requires the torpedo remain within a 3,000 foot radius of the center of the hydrophone array.

Figures 6 and 7 depict the error in both position and depth for the straight run trajectory along the X axis. The position errors are computed by subtracting the filter

position estimate, $x(k/k)$, from the computer generated true position for each time slot. From Figure 6, it is observed that the Y position error magnitude, which is caused by system noise, decreases on the inbound leg and increases on the outbound leg. Since the torpedo path originated on the outer tracking perimeter ($X = 3,000$ feet), the noise in the Y channel will be amplified due to the large uncertainty in the Y position estimate. The magnitude of noise error in the Y direction also decreases as the torpedo approaches the array and increases as the torpedo moves outbound. The steady state X and Y position errors ranged between -11 and +4 feet throughout the trajectory.

Figure 8 describes a straight inbound torpedo path that passes north of the array at a constant Y distance of 2121 feet. Torpedo velocity is maintained at 20 knots. A comparison of the RMS x estimates of this track with the preceding one, Figures 5 and 9, reveal that both estimates are relatively constant, however the X estimate in Figure 9 is significantly larger. Since the torpedo path is parallel to the X axis, a larger uncertainty in the X position estimate will result. A comparison of the RMS Y position error estimates reveals identical propagation characteristics. This is due to the symmetry in the east-west direction of the two trajectories.

Figures 10 and 11 show that the steady state position and depth magnitude errors are less than 10 feet. The noise

induced error in the X position is due to the uncertainty in the filter X position estimate.

A torpedo path which is inbound along a 45° diagonal to the array is depicted in Figure 12. For this run torpedo velocity is constant at 28 knots. Figure 13 shows that the filter's X and Y RMS estimates of position error are approximately equal. Likewise Figure 14 shows that X and Y position error propagation is also similar. These results are consistent since the X and Y position magnitudes of the torpedo trajectory are identical. At the point of closest approach ($X = 0$, $Y = 0$) the uncertainty in both the X and Y positions becomes a minimum and increases as the torpedo moves past the array. Steady state position and depth errors range between -4 and +6 feet.

B. SERIES TWO--MANEUVERING RUNS

The results of the straight run analyses discussed in the previous subsection show that the propagation of the filter's estimate of RMS position error is dependent on the path of the torpedo with respect to the hydrophone array. Thus the errors in X and Y position, and Z depth are also a function of the torpedo trajectory. In this series of tests maneuvers are added to the three straight path trajectories. These maneuvers consist of 360° turns at turn rates of 10 and $20^\circ/\text{sec}$. G forces range between $1/4\text{-G}$ and 1-G . G forces are changed using the same turn rate by varying velocity. A typical test

run on the tracking range at Dabob would include these type maneuvers.

Figure 16 describes an inbound torpedo path which maintains a Y distance from the array between -200 and +600 feet. Initial position and velocity are 30 and 20 feet in X and Y position and 30 and -20 feet/second in X and Y velocity. Torpedo velocity is maintained constant at 20 knots throughout the trajectory. Four 360° $1/4$ -G turns are executed during the run. Figures 17 and 18 describe the position and depth errors. Within approximately three time slots the filter achieved steady state performance. Upon observing the error propagation it is apparent that the filter is insensitive to turn rate maneuvers. Negligible degradation in position and depth is observed. Thus the error bound routine discussed in the previous section successfully bounds position and depth errors during turns. As expected the position errors decrease as the torpedo approaches the array and increase as the torpedo proceeds outbound from the array. Since position and depth errors exhibit approximately equal oscillations about zero indicates that the system noise is the dominant error source driving the filter. Bias errors on the time residuals due to maneuvers are negligible. Steady state errors in position and depth ranged between -1 and +5 feet throughout the run.

The next test describes an inbound torpedo trajectory, Figure 19, which passes north of the hydrophone array. The torpedo maintained a Y distance from the array between 1600

and 2400 feet. Four $10^\circ/\text{sec}$ 1/4-G turns are executed during this test. Filter initialization errors are 20 and 11 feet in X and Y position and 20 feet/sec in both X and Y velocity. The error in position and depth is described in Figures 20 and 21. The X position error demonstrates larger deviations than the Y position error. As noted during the straight runs, this is due to the difference in the filter's X and Y RMS position error estimates for this type trajectory. The oscillatory characteristics of the position error indicate again that noise is the dominant error source and the filter is tracking the torpedo through the maneuvers. Position errors range between -8 and +8 feet with one X channel error reaching a maximum of -15 feet, which occurred at time slot 140. The exact cause of the -15 foot error is unknown but may have been caused by an anomaly in the IBM-360 subroutine SNORM which generates the zero mean Gaussian noise for the filter. Errors in torpedo depth, Figure 21, are consistent and range between -2 and +5 feet.

For the next test the torpedo exhibits an inbound trajectory along a 45° diagonal with respect to the array. Figure 22 shows that four $10^\circ/\text{sec}$ 1/4-G turns are executed on this run: 2 on the inbound leg and 2 on the outbound leg. Filter initialization errors are 60 feet in both X and Y position and 30 feet in X and Y velocity. Steady state conditions are achieved in approximately four time slots. As expected for this type trajectory, the X and Y position

errors portray similar propagation characteristics, Figure 23. This similarity can be explained by the equivalent X and Y RMS error estimates generated by the filter for a 45° diagonally inbound trajectory. Position and depth errors, Figure 24, ranged between -7 and +10 feet.

The preceeding tests discussed filter performance for straight runs with 10°/sec 1/4-G turns. Using the same basic torpedo trajectories as above, similar tests are performed with 10°/sec turn rates at 1/2-G. Figures 25 through 33 describe the torpedo paths and the corresponding position and depth error propagation plots for each run. Filter initialization errors ranged between 10 and 40 feet in position and 10 and 40 feet in velocity. Torpedo velocity is maintained constant at 55 knots. An analysis of the position and depth plots reveals that filter performance is similar to that described in the previous tests. In Figures 26 and 32 the larger errors incurred after time slot 135 are due to the torpedo trajectory exceeding the radial tracking range limit of 3,000 feet. This is more clearly described in the torpedo trajectory curves, Figures 25 and 31. The additional 1/4-G acceleration during the turns did not produce visible degradation in the filter's position and depth estimates.

The next set of maneuvering tests increased the turn rate to 20°/sec with G forces at 1/2-G. The same basic inbound and outbound straight runs trajectories are used. Initialization errors ranged between 20 and 40 feet in position and -10 and

and 35 feet/second for velocity. Figure 34 describes a torpedo inbound trajectory which maintains a Y distance from the array between -275 and 500 feet. Four turns are executed on this run. The X and Y position errors reveal essentially the same characteristics described before. In general the position and depth errors decreased as the torpedo approaches the array, become a minimum at the closest point of approach, and increase as the torpedo moves outbound past the array. The higher turn rates experienced on this test do not degrade the X and Y position errors. Figures 35 and 36 show that position and depth errors ranged between -1 and +5 feet.

Figure 37 depicts a torpedo path which maintains a Y distance from the array between 1760 and 2760 feet. Three $20^\circ/\text{sec}$ $1/2\text{-G}$ turns are performed during this run. Figures 38 and 39 describe the position and depth errors which range between -7 and +9 feet during steady state.

The diagonally inbound trajectory with four $20^\circ/\text{sec}$ $1/2\text{-G}$ turns is displayed in Figure 40. Position and depth errors for this run, Figures 41 and 42, ranged between +10 and -10 feet. Again note the similarity in the position error behavior for each channel for this type trajectory. The larger errors in position after time slot 125 are due to the filter's RMS position error estimates increasing as the torpedo approaches the radial tracking limit of the hydrophone array.

The last set of maneuvering tests increased the G force during the $20^\circ/\text{sec}$ turns from $1/2\text{-G}$ to 1-G . The three basic

inbound-outbound trajectories are again repeated. Figures 43 through 51 describe the torpedo trajectories and corresponding position and depth errors. Overall performance indicates that during steady state conditions the magnitude of position and depth errors are below 10 feet.

Due to the presence of system noise position errors reached a maximum of 14 feet, when the torpedo trajectory exceeded the radial tracking limit of the hydrophone array. As was also observed in the preceding tests, errors induced by the turn rate appeared to have a negligible effect on filter performance.

C. SERIES THREE--MULTIPLE ARRAY TRACKING

The purpose of the last series of tests is to functionally demonstrate that the filter is able to track the torpedo through multiple arrays using the simulated tracking range set up and handoff scheme described in the previous section. Figure 52 depicts a straight run trajectory which traverses the six hydrophone arrays. The position and depth errors are described in Figures 53 and 54.

Note that the handoff from array to array was accomplished with negligible effect on position and depth errors. The magnitude of the errors compare well with those observed on the single array straight run tracking.

VII. CONCLUSIONS

The sequential Extended Kalman filter satisfactorily provided real time estimates of torpedo position and depth. The magnitude of steady state position and depth errors ranged between 0 and 10 feet for torpedo tracks within the specified radial tracking range.

The iterative error bound routine furnished the capability to estimate torpedo position and depth during turning maneuvers up to 1-G. Errors generated during maneuvers are comparable to those observed during straight runs.

The filter's RMS estimates of position and depth errors are dependent on system noise and the distance the torpedo is from the center of the hydrophone array. In general larger errors are encountered at more remote distances from the hydrophone array.

The simulated trajectories were performed using constant measurement noise sequences. Since the filtering problem is non-linear the error, in the filter's state estimate vector is a function of the statistical properties of the noise. Future tests should include a Monte Carlo simulation where filter performance is investigated for an ensemble of measurement noise sequences.

Additional tests should also include evaluating filter performance using trajectories generated from actual torpedo

runs on the Dabob test range. These tests would verify the adequacy of the noise model in the filter and the ability of the software to edit erroneous transit time measurements.

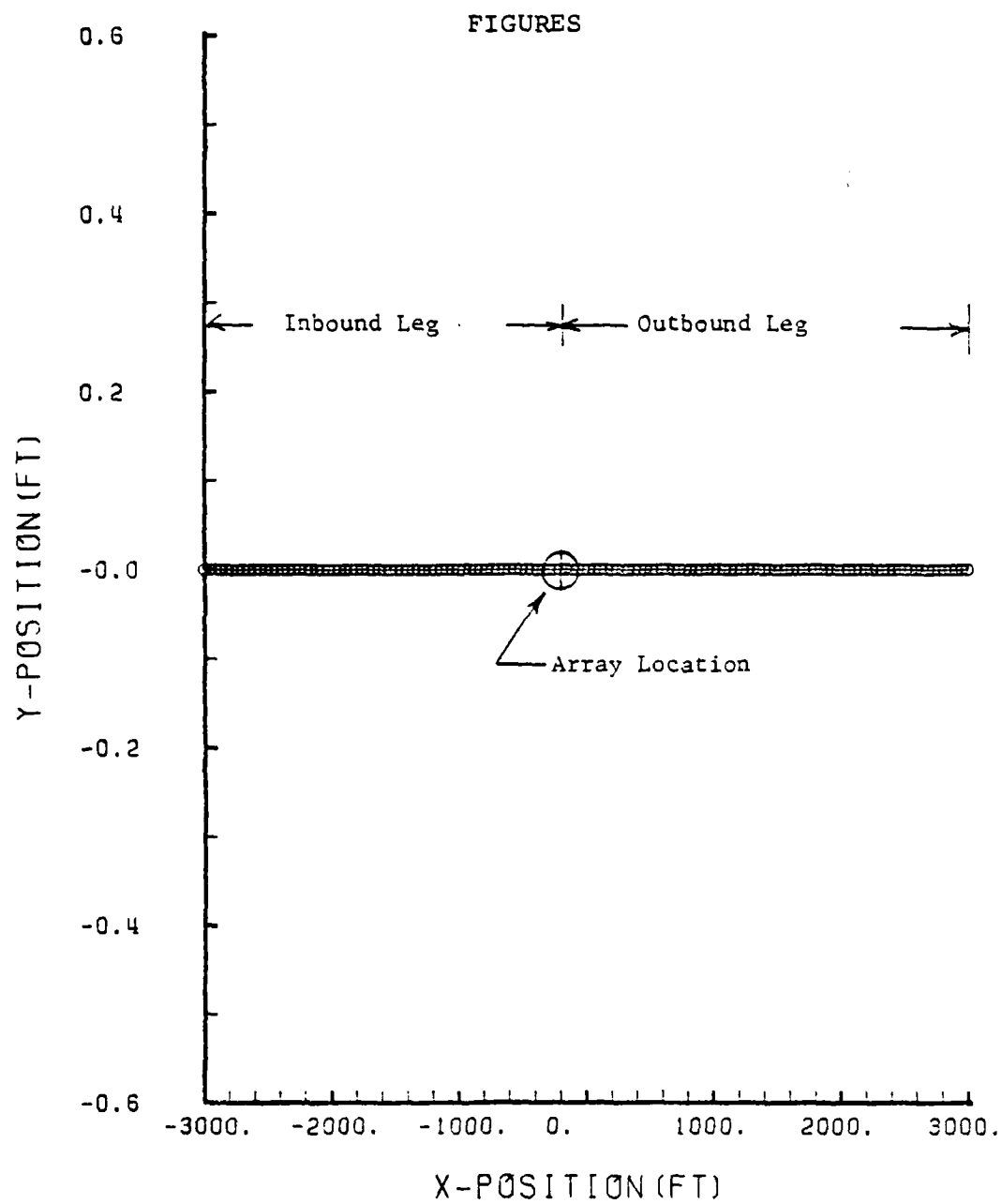


Figure 4. True Trajectory of the Torpedo in the Area of a Single Array

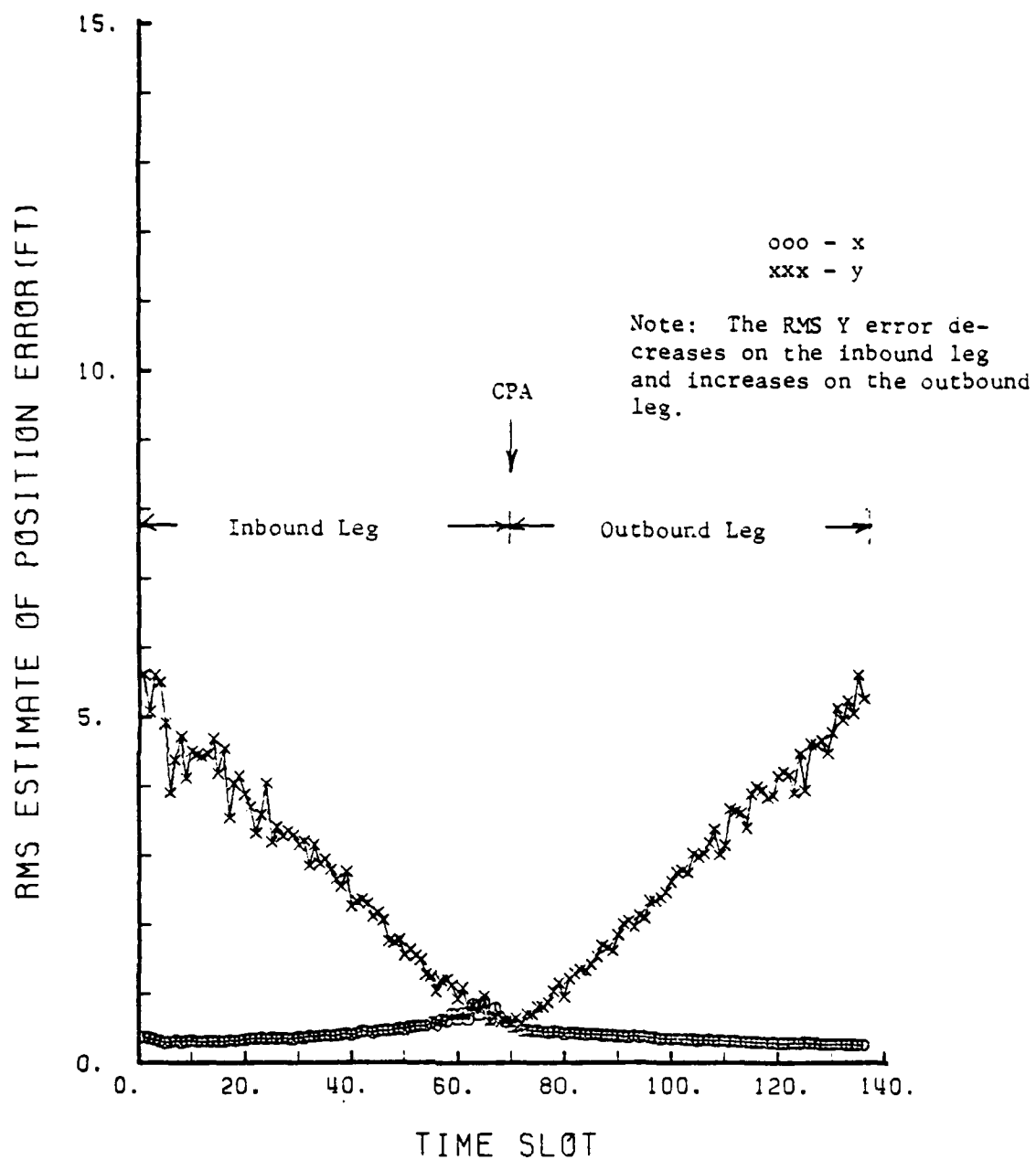


Figure 5. Filter Estimate of RMS Position Error During a Straight Run in the Area of a Single Array

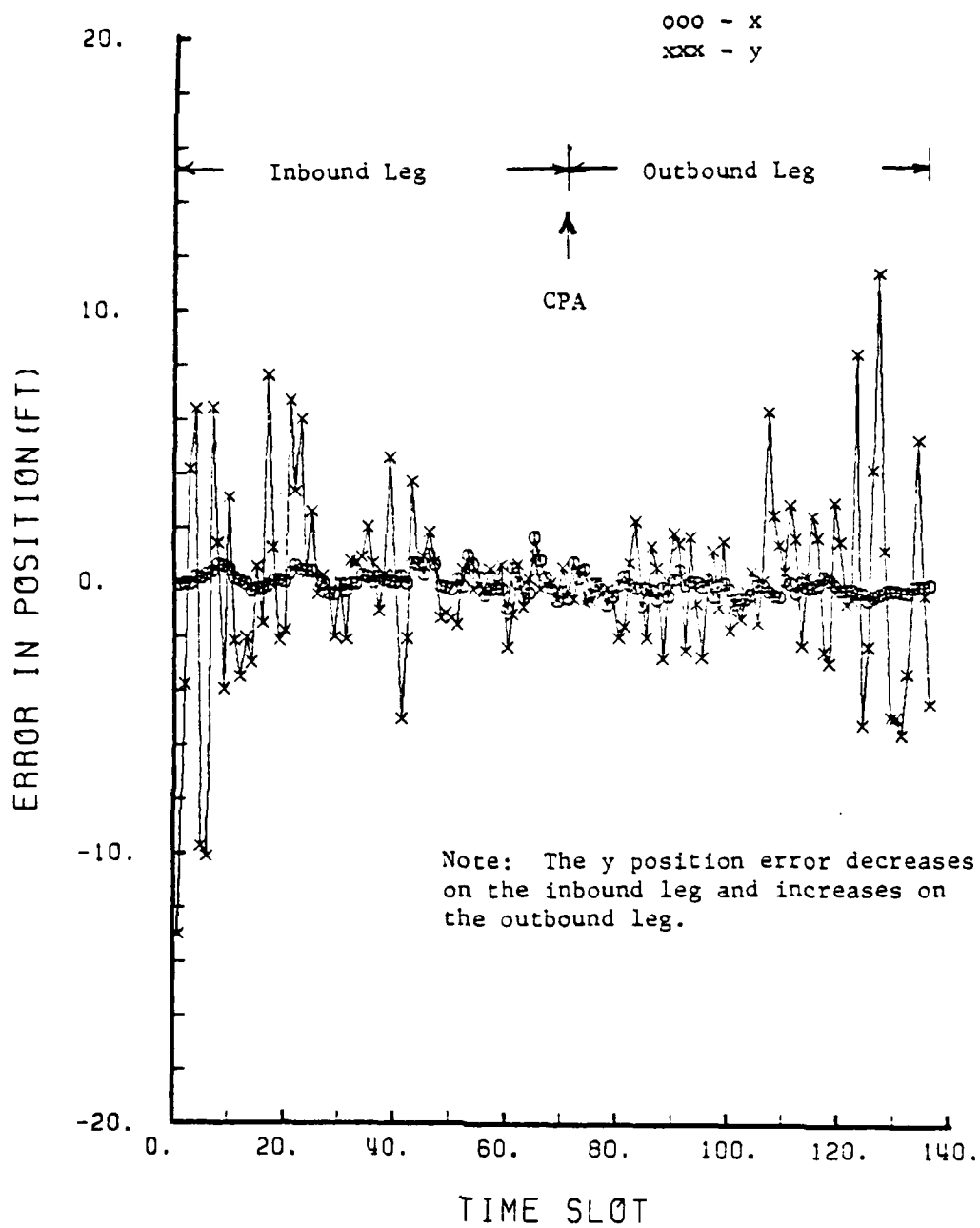


Figure 6. Error in Torpedo Position During A Straight Run in the Area of a Single Array

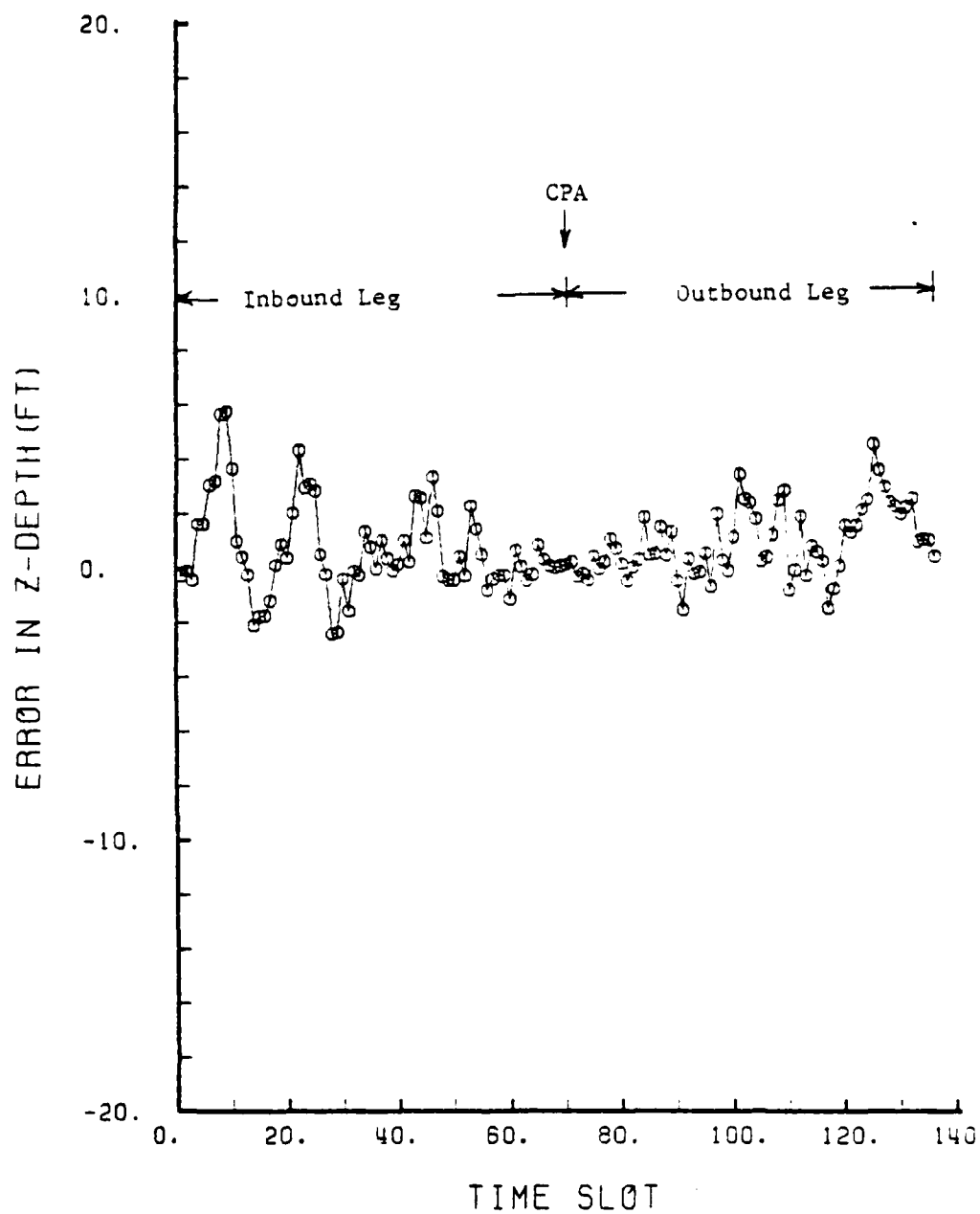


Figure 7. Error in Torpedo Depth During a Straight Run in the Area of a Single Array

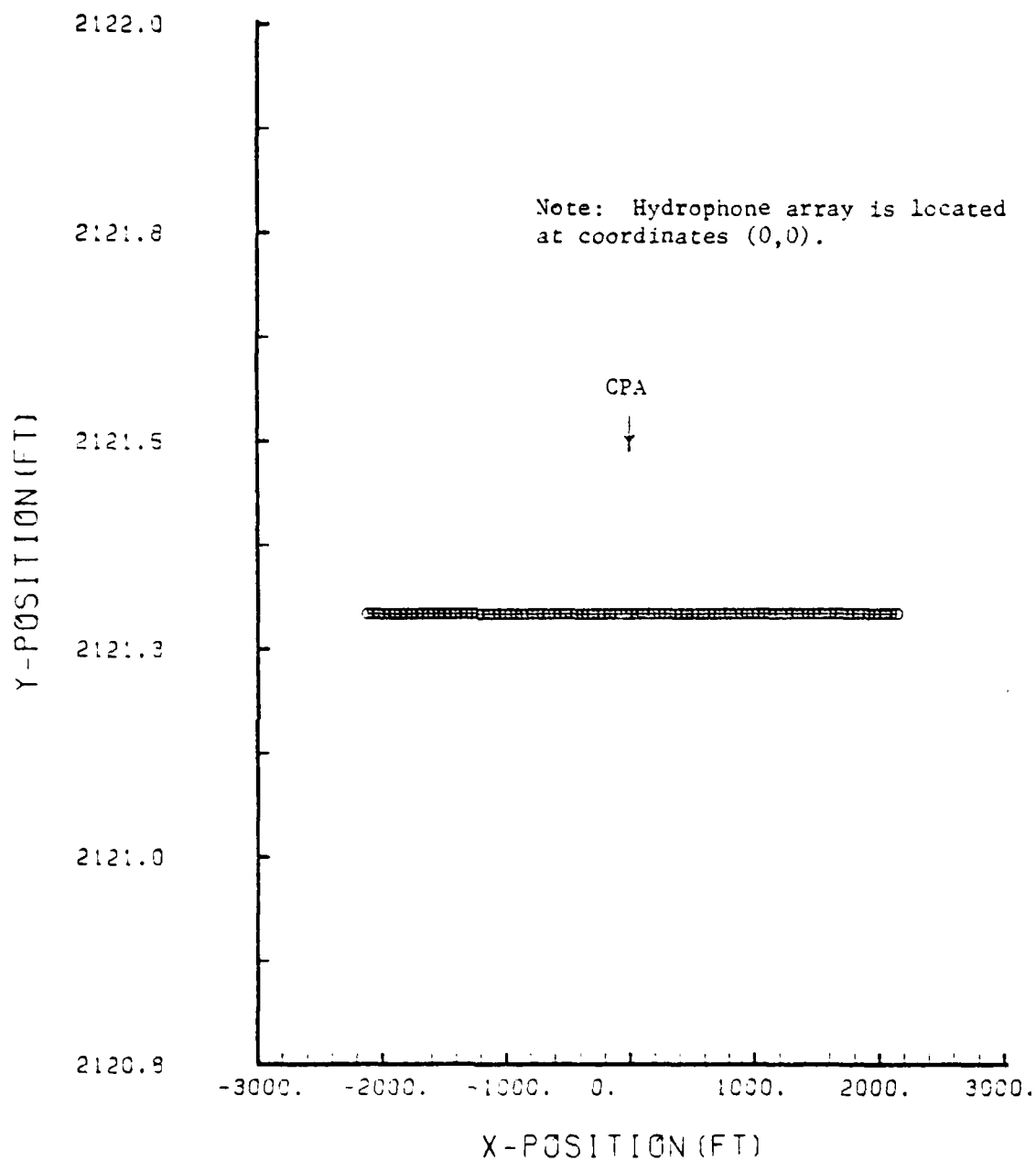


Figure 8. True Trajectory of the Torpedo During a Straight Run in the Area of a Single Array.

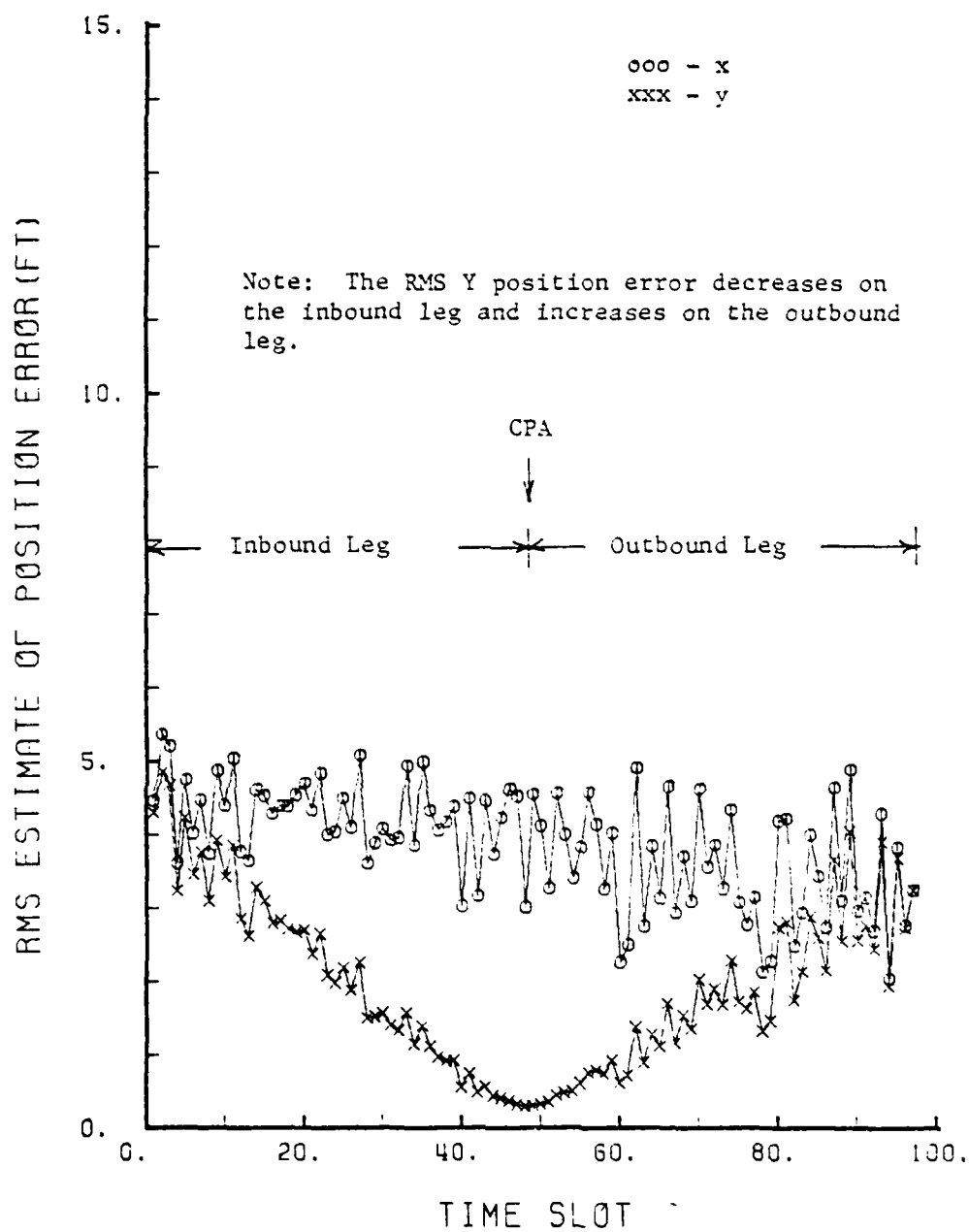


Figure 9.. Filter Estimate of the RMS Position Error During a Straight Run in the Area of a Single Array

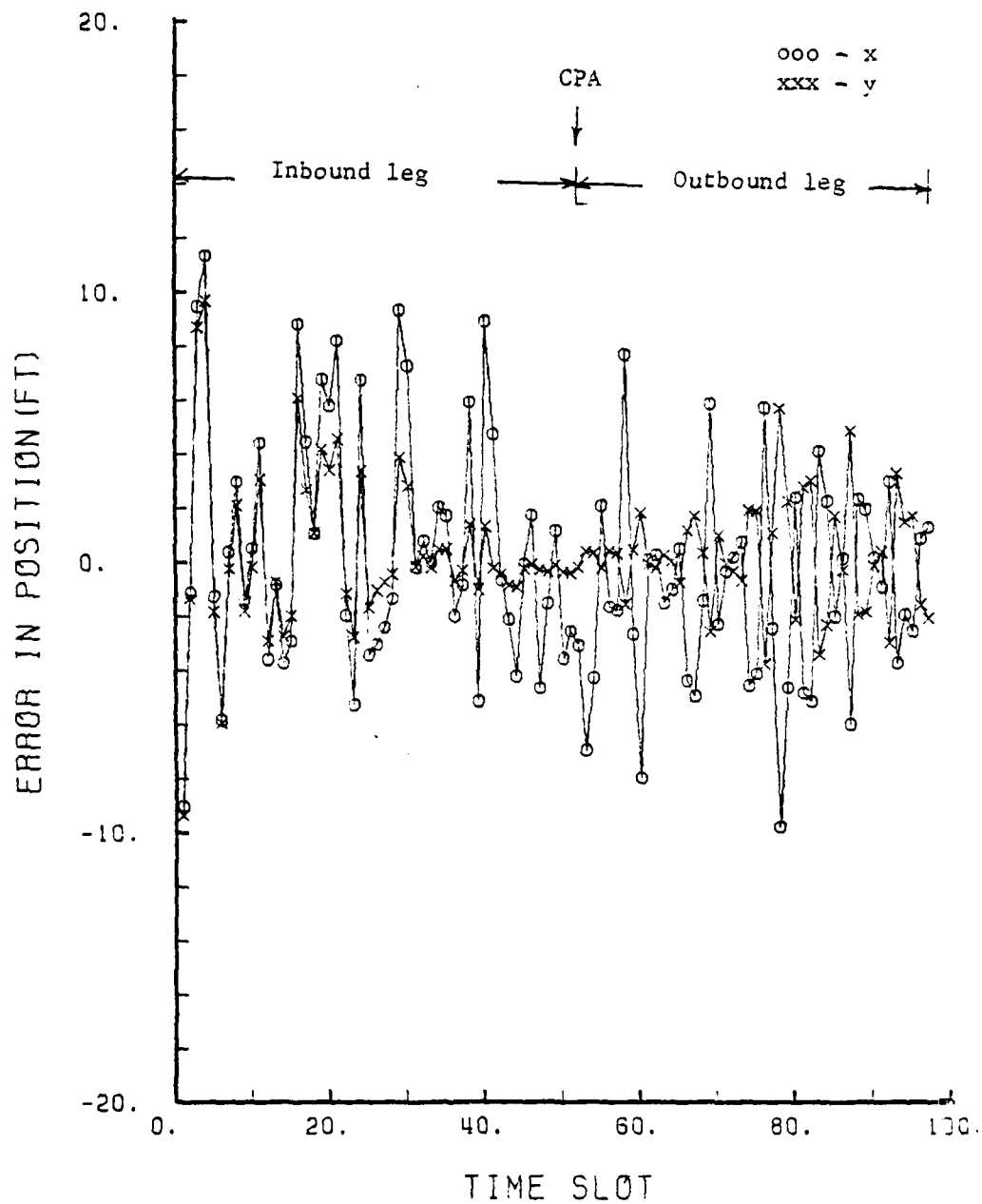


Figure 10. Error in Torpedo Position During a Straight Run in the Area of a Single Array

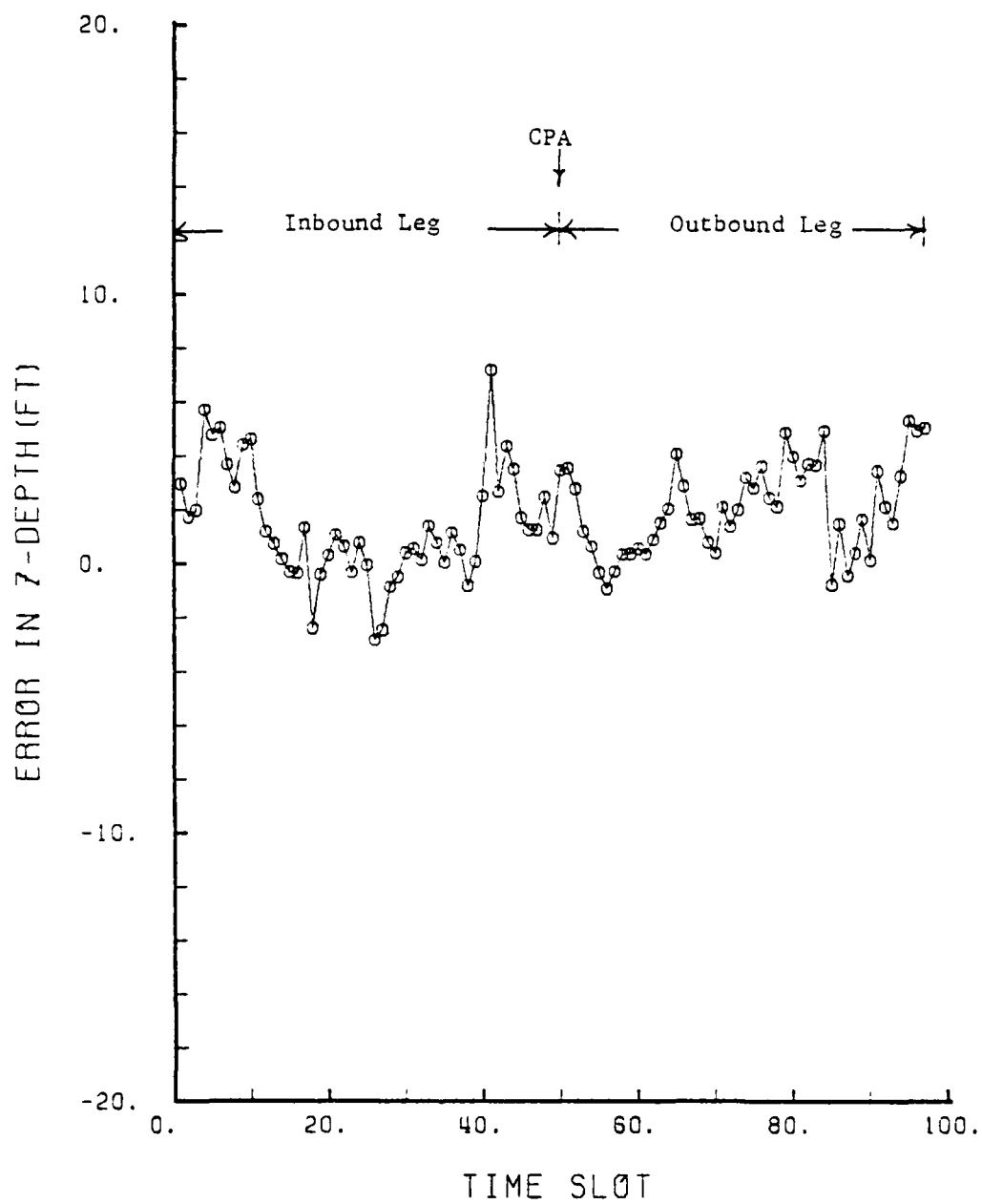


Figure 11. Error in Torpedo Depth During a Straight Run in the Area of a Single Array

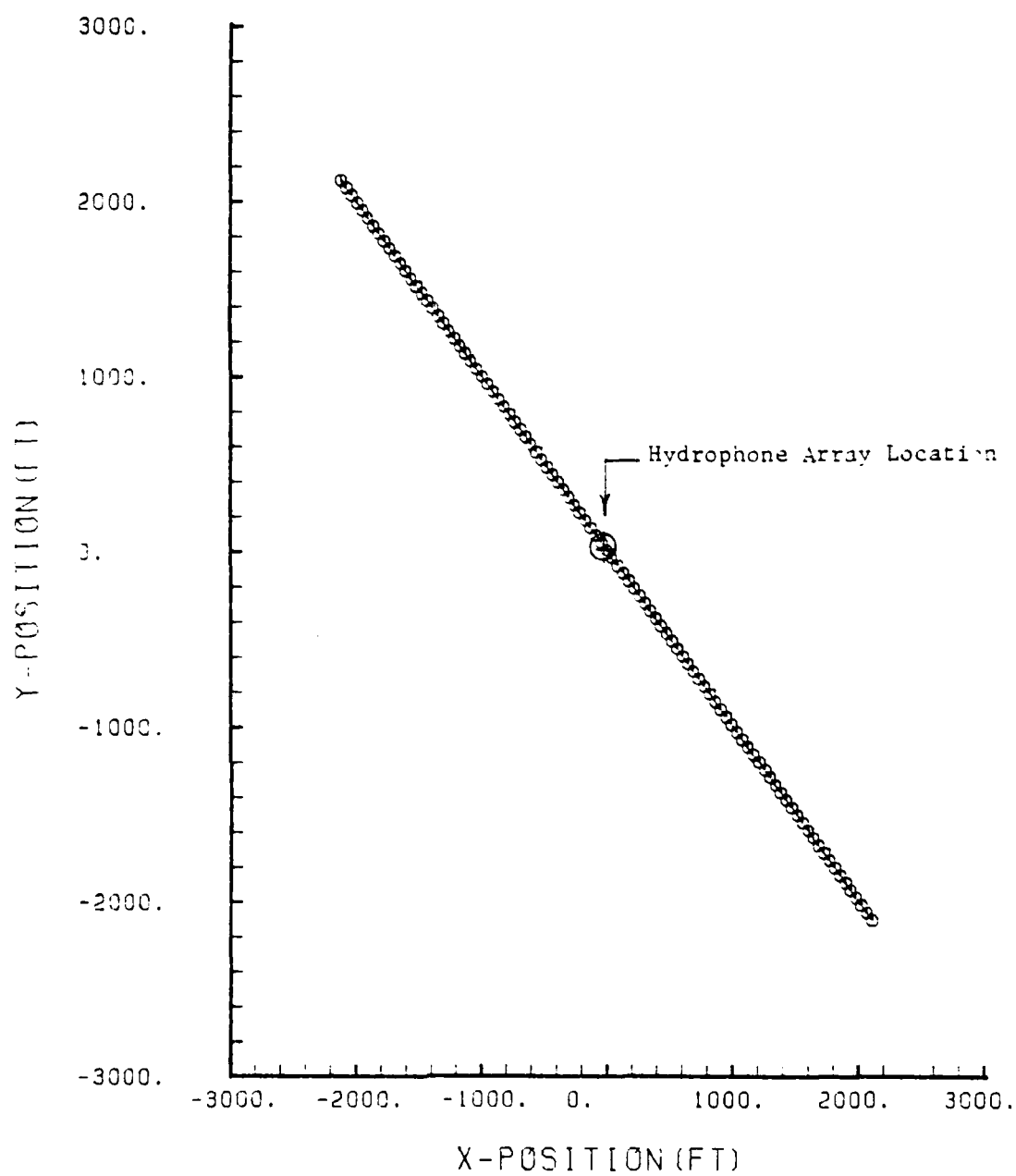


Figure 12. True Trajectory of the Torpedo During a Straight Run in the Area of a Single Array

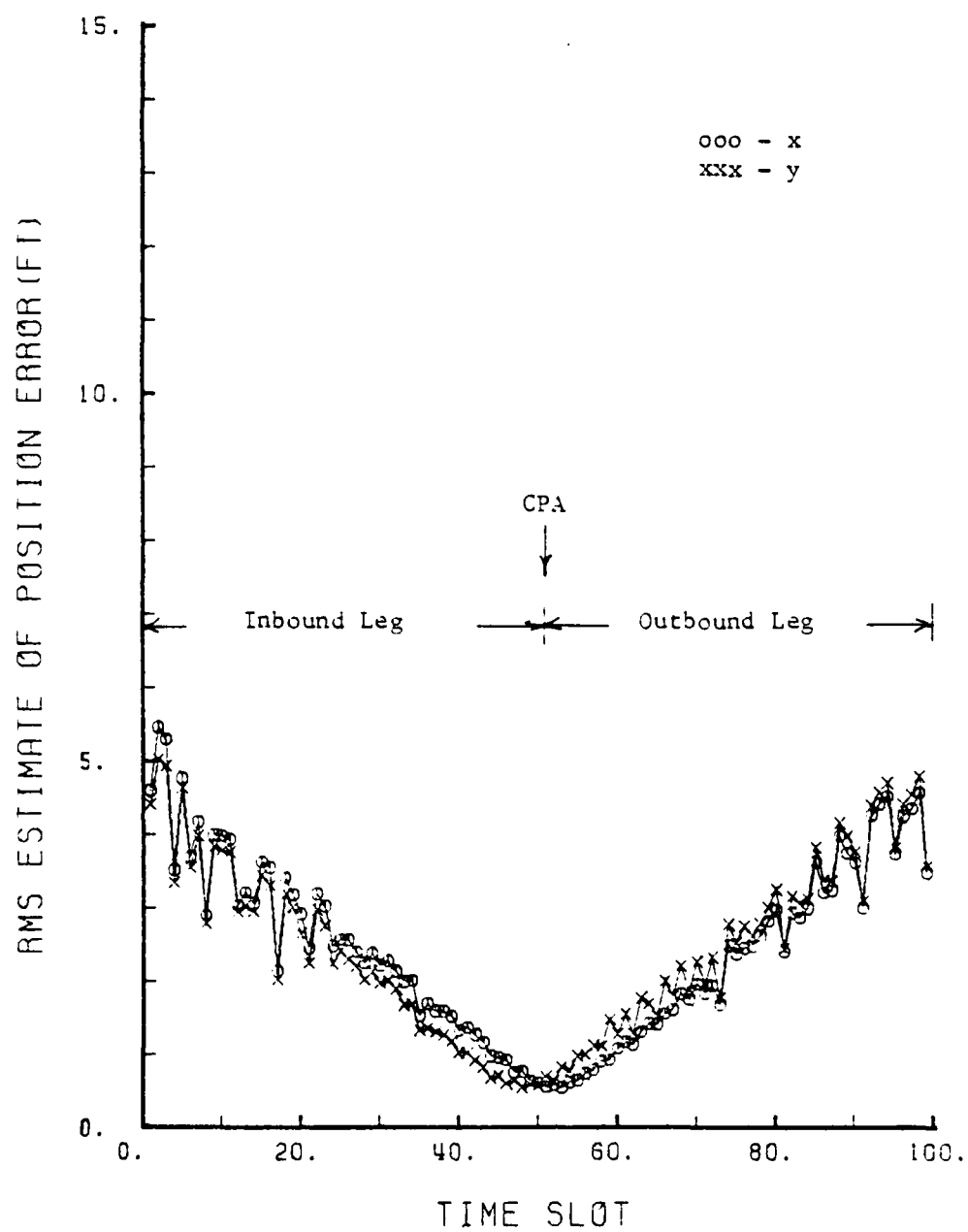


Figure 13. Filter Estimate of RMS Position Position Error During a Single Run in the Area of Single Array

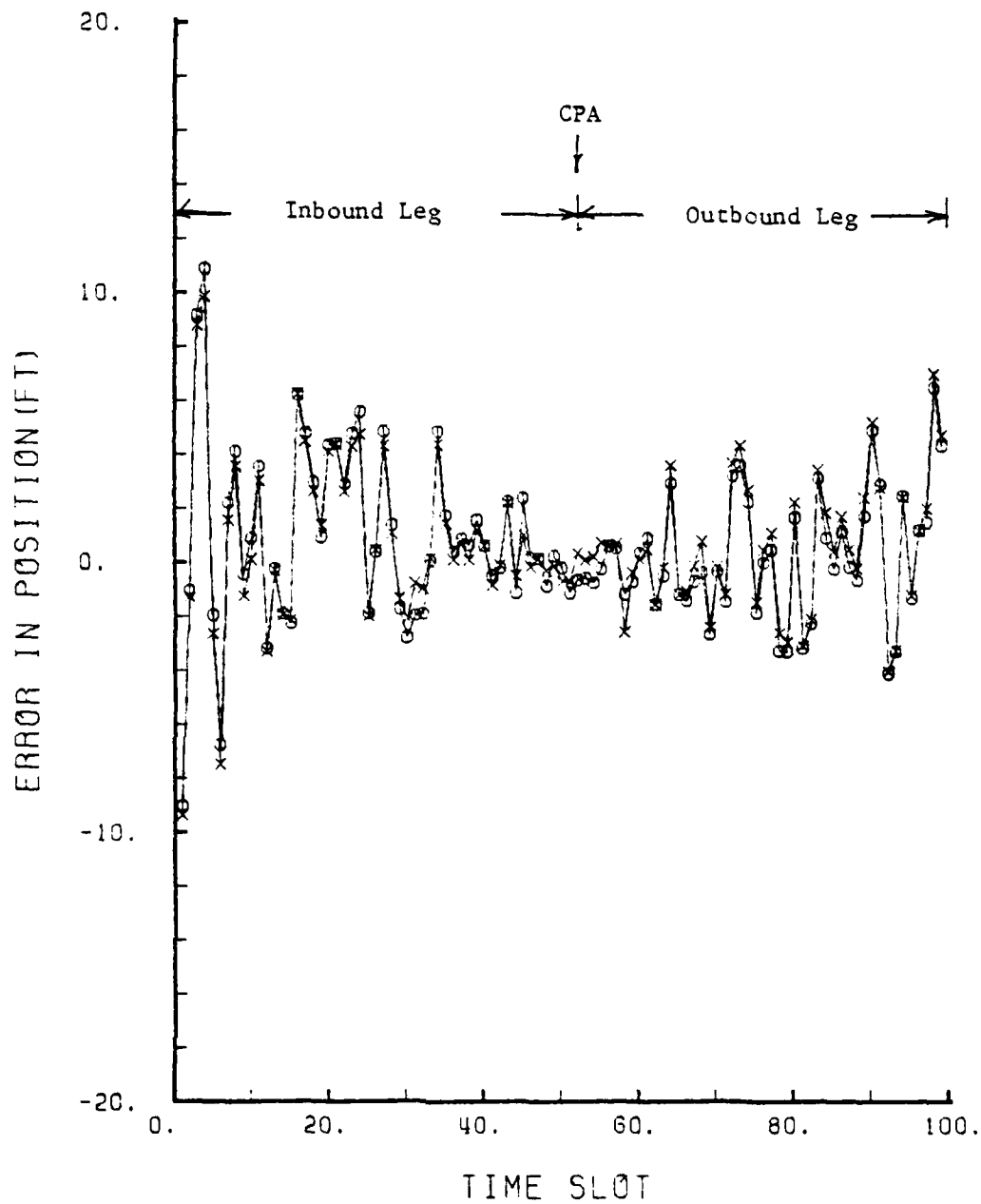


Figure 14. Error in Torpedo Position During a Straight Run in the Area of a Single Array

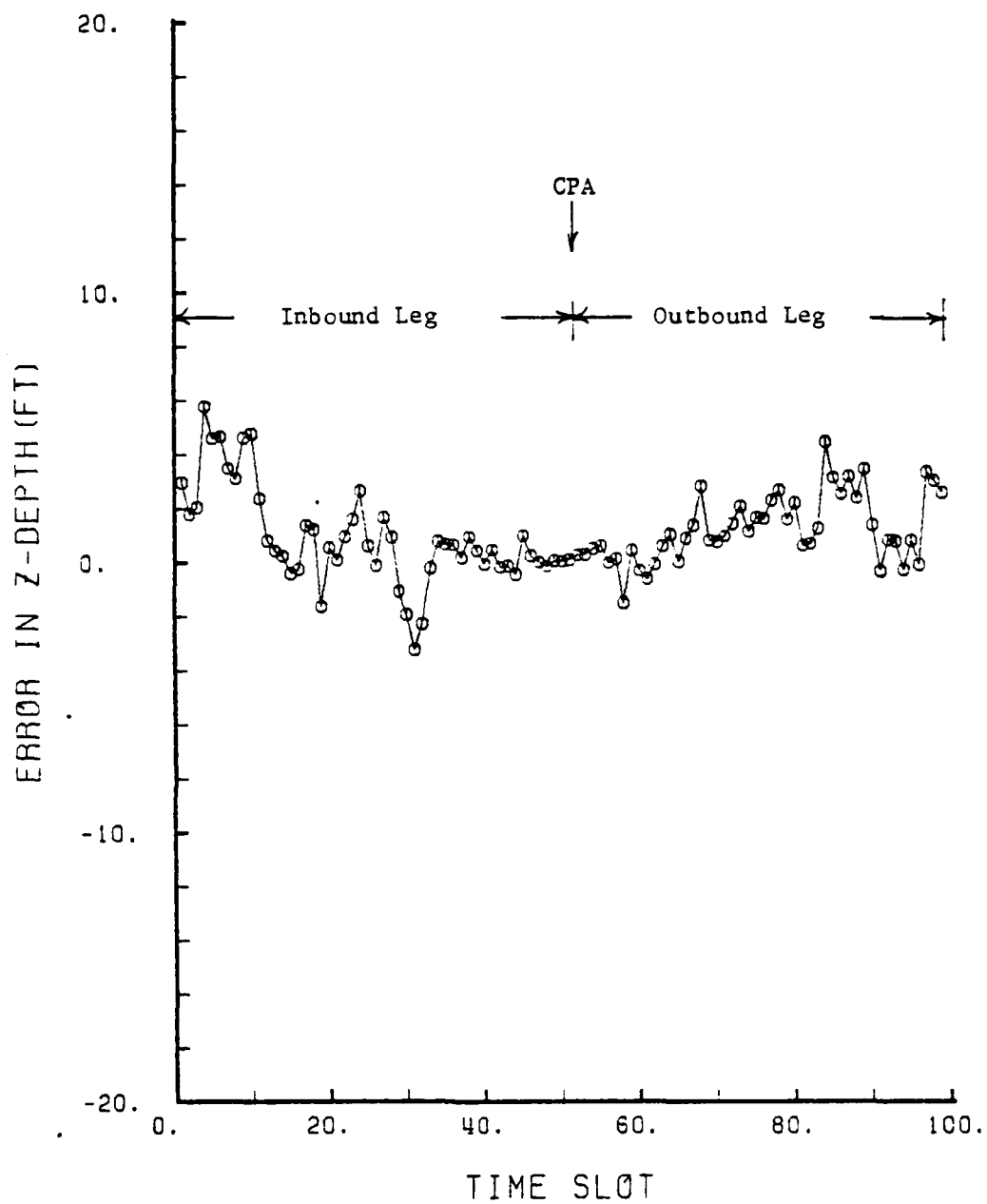


Figure 15. Error in Torpedo Depth During a Straight Run in the Area of a Single Array

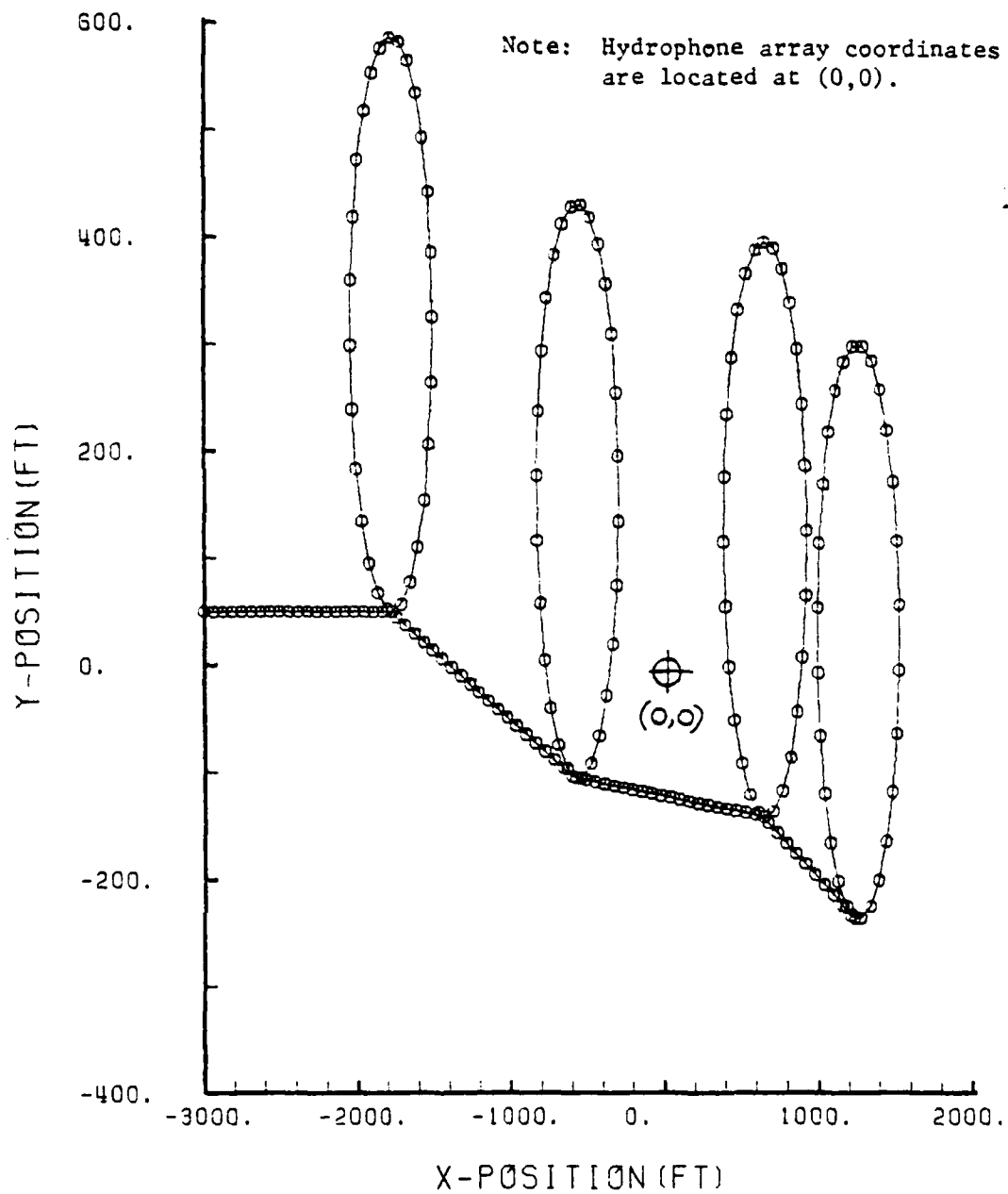


Figure 16. True Trajectory of the Torpedo in the Area of a Single Array

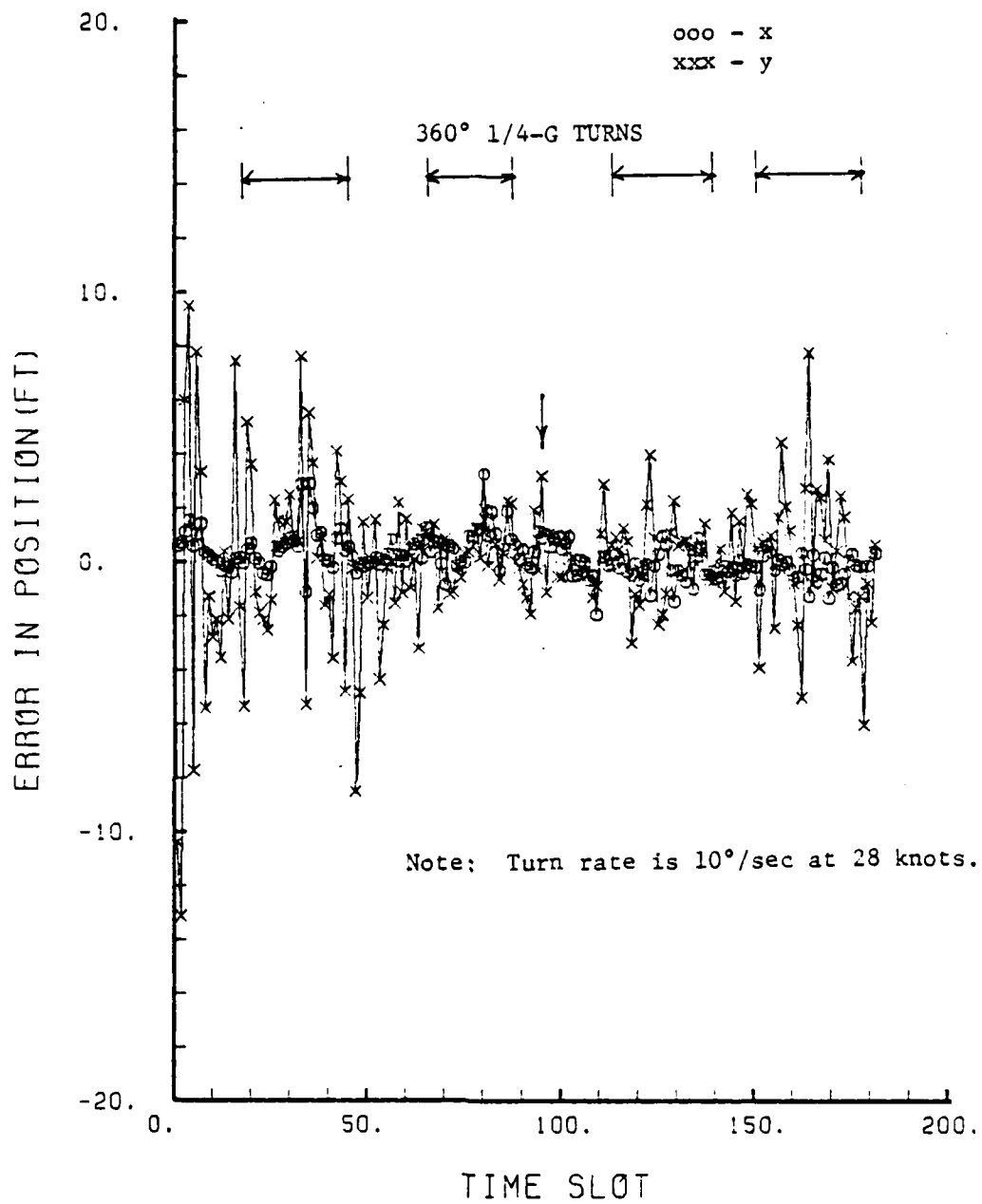


Figure 17. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

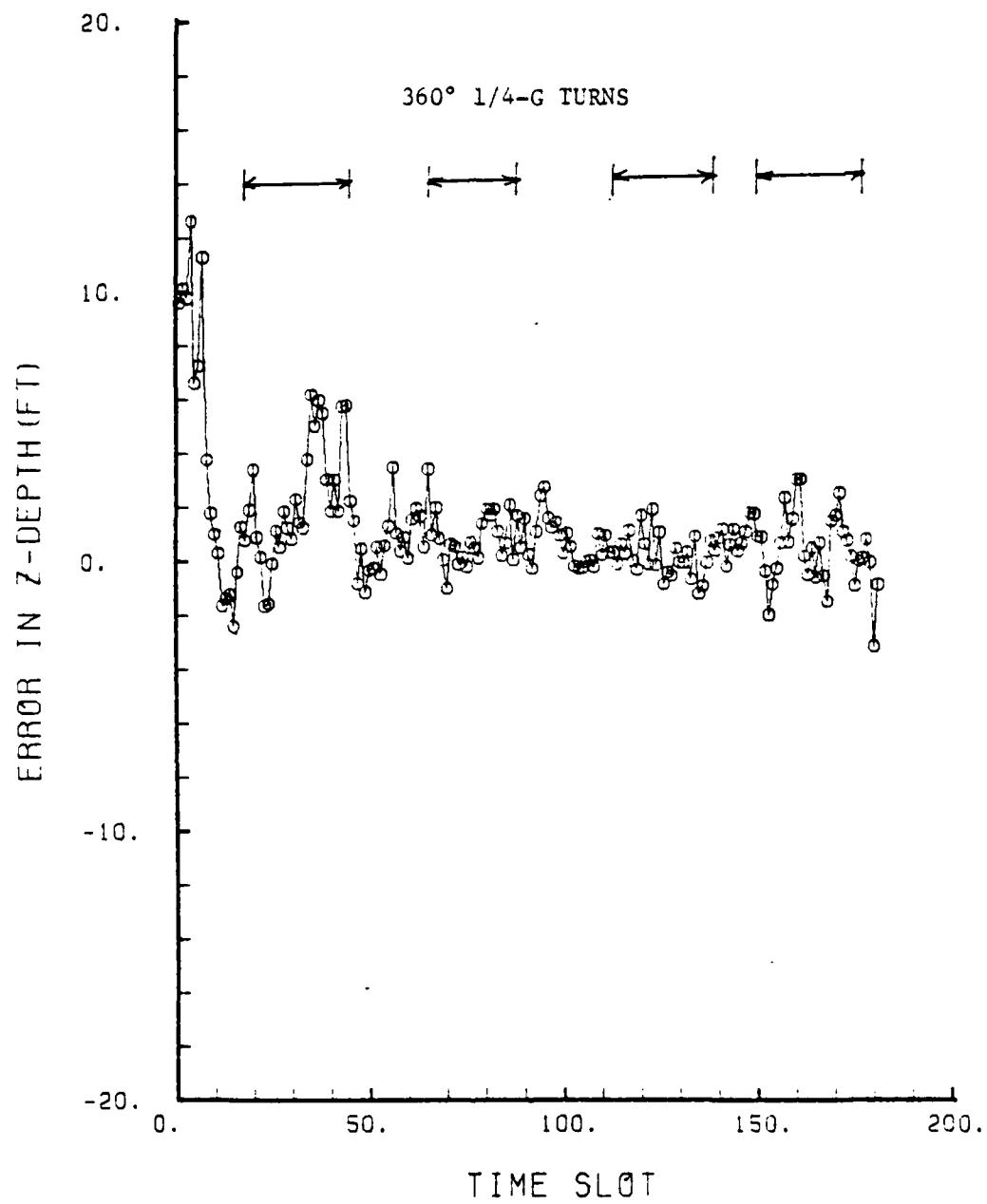


Figure 18. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

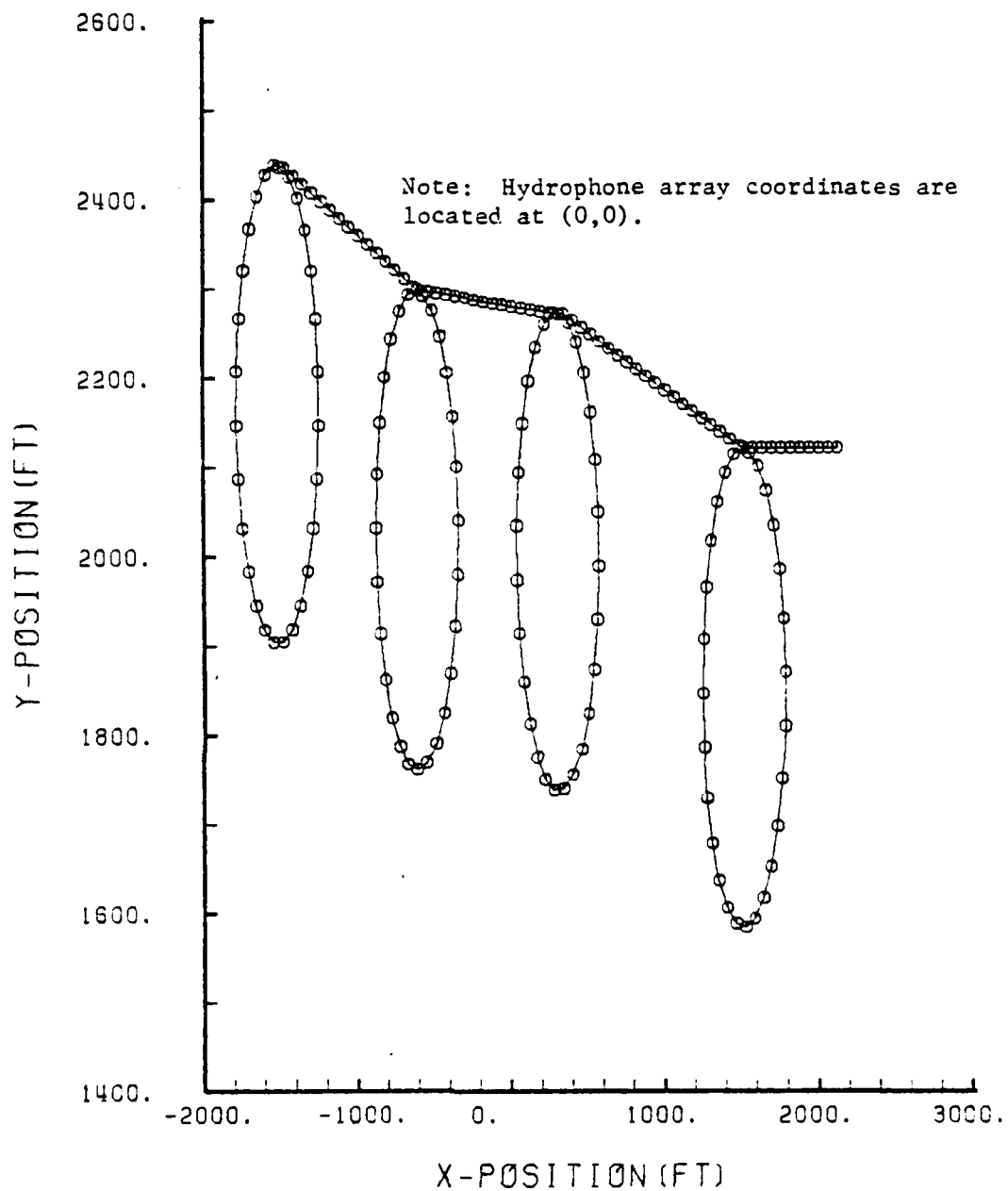


Figure 19. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

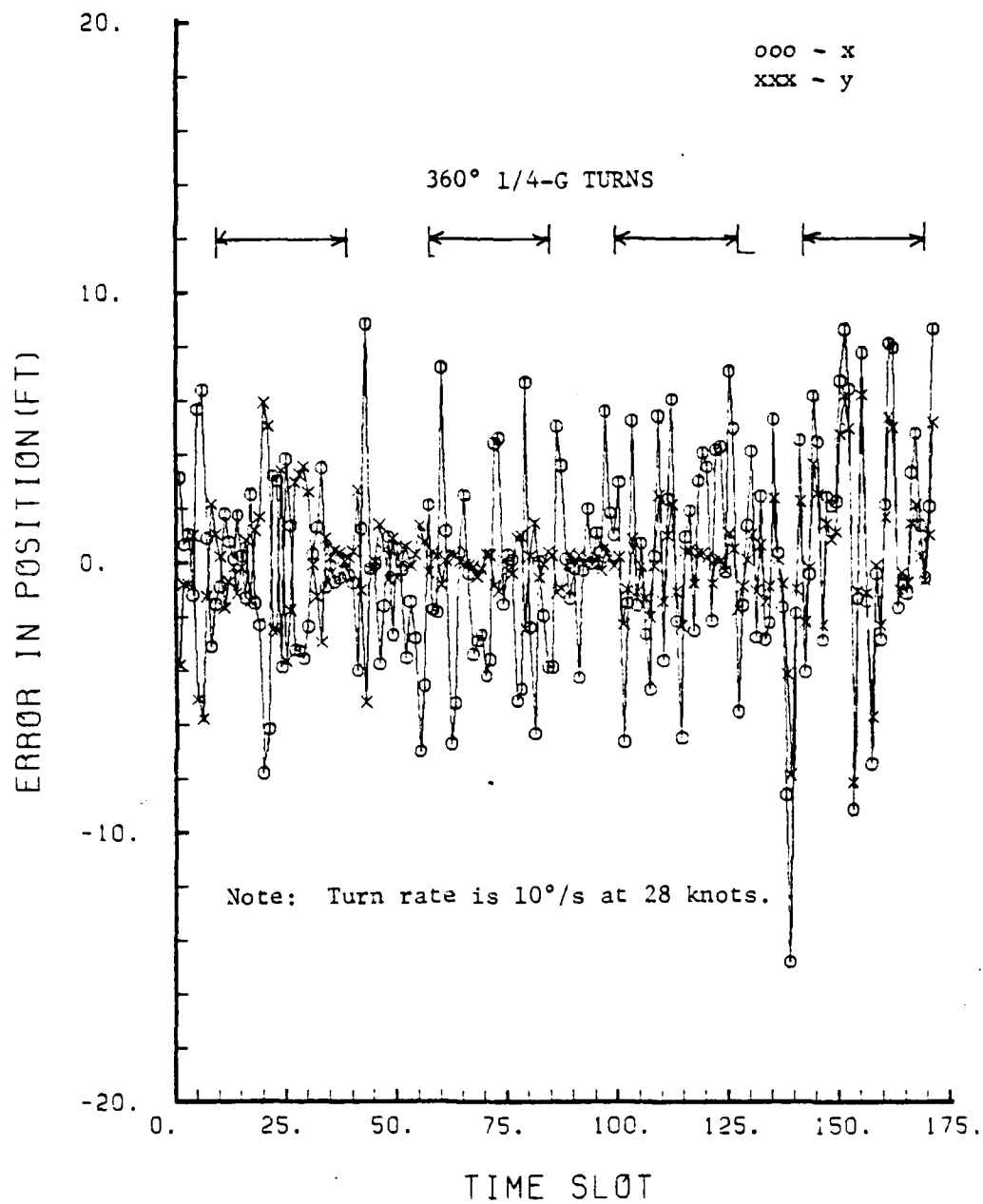


Figure 20. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

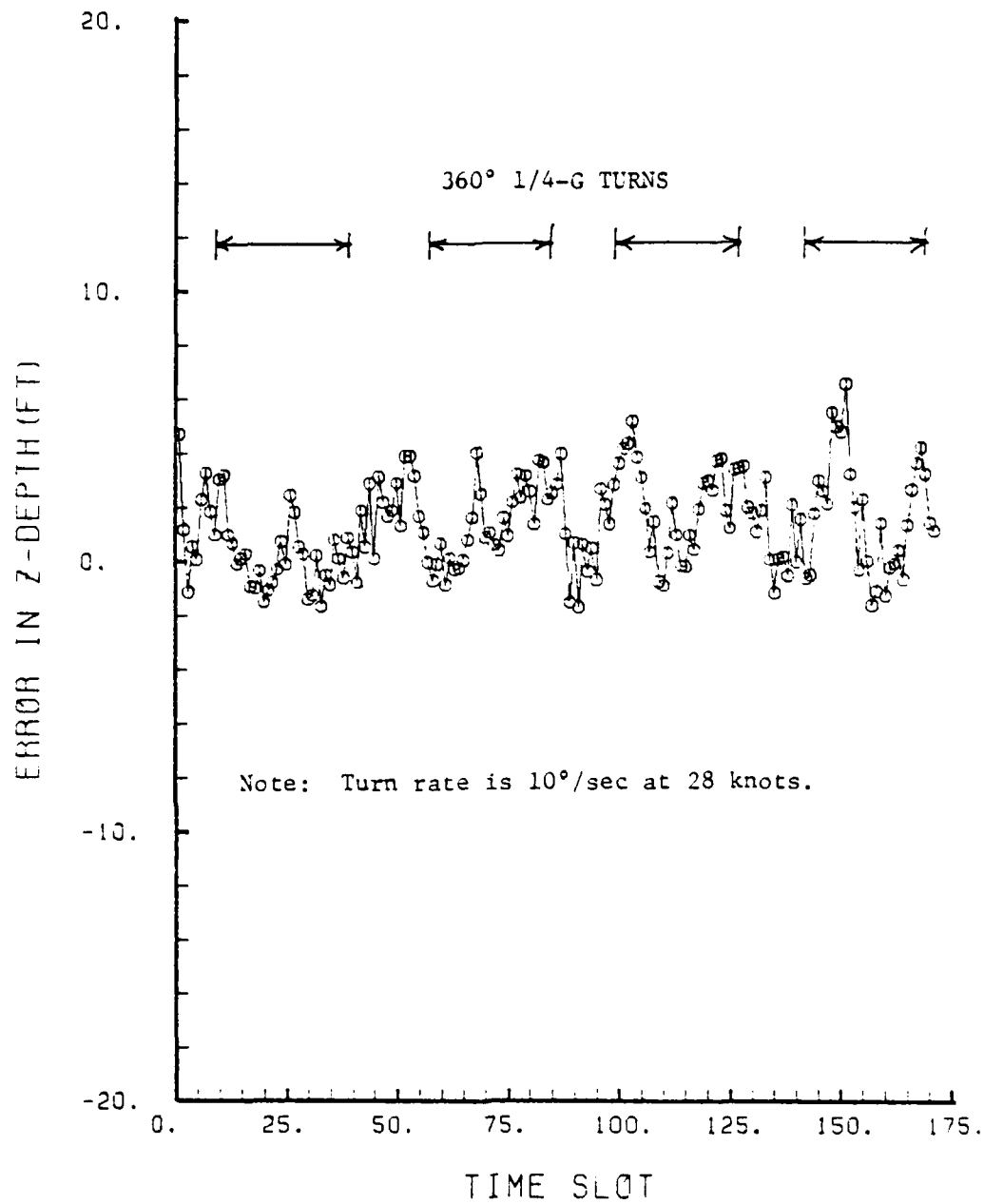


Figure 21. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

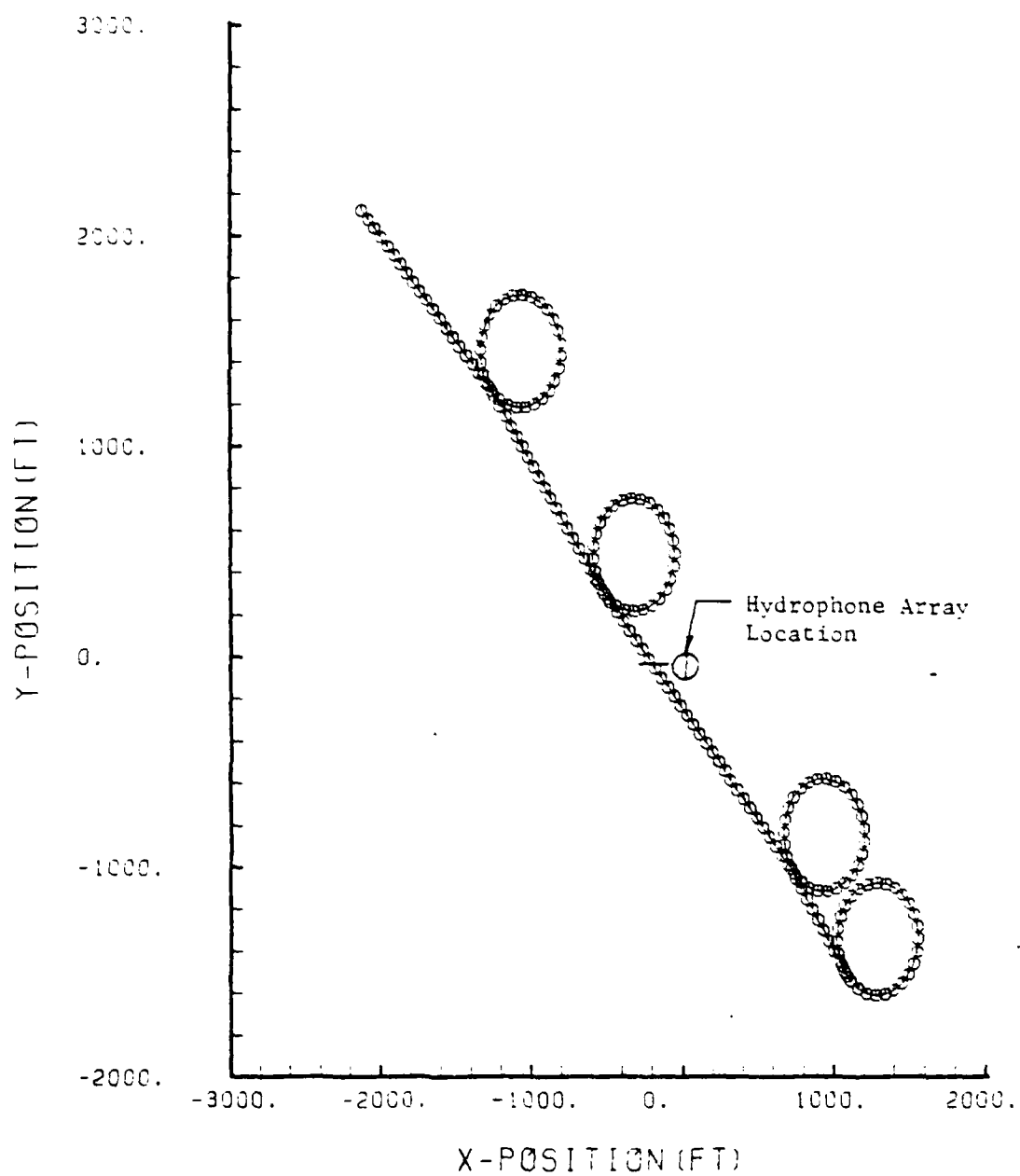


Figure 22. True Trajectory of the Torpedo During a Maneuvering Run in the Area of A Single Array

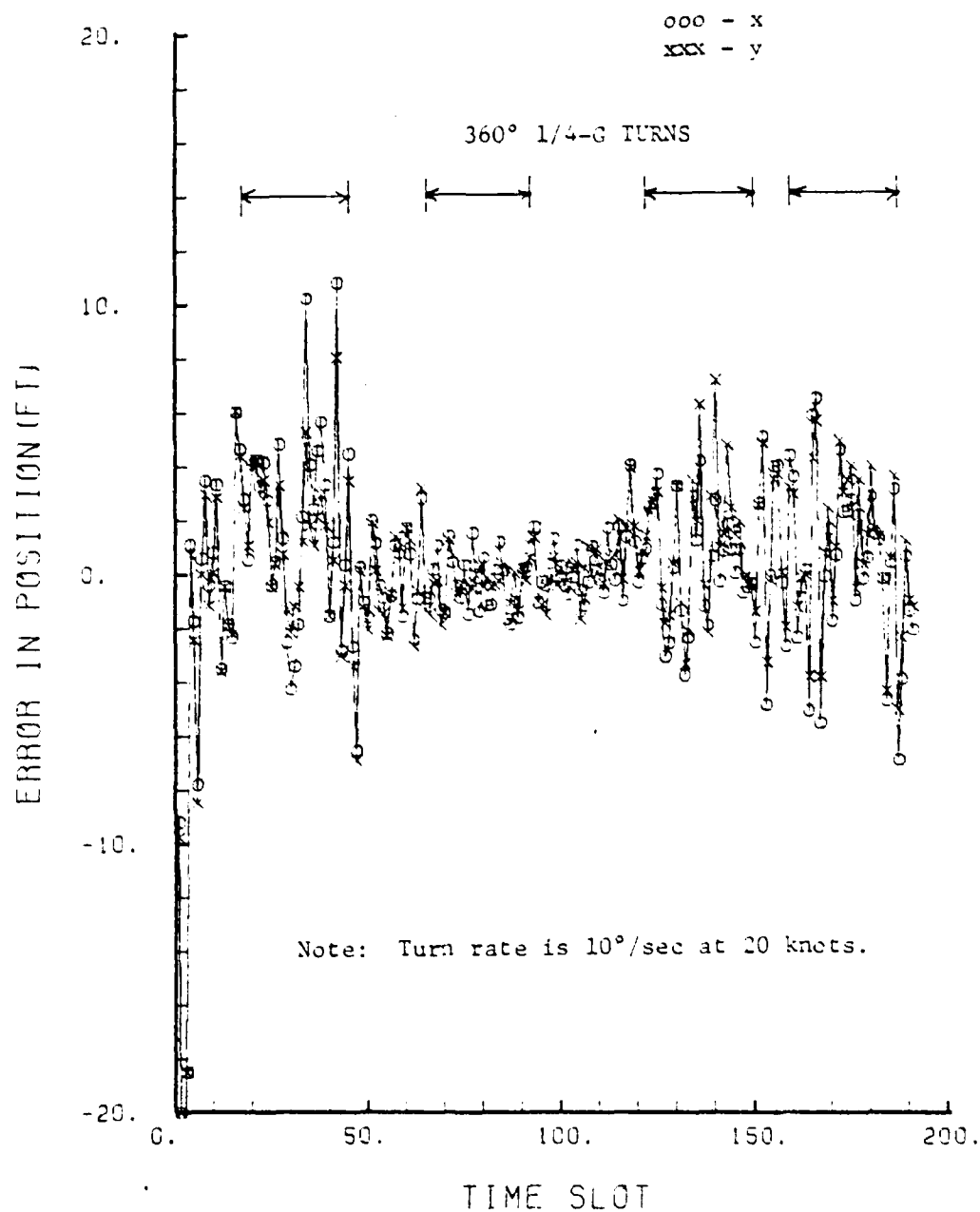


Figure 23. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

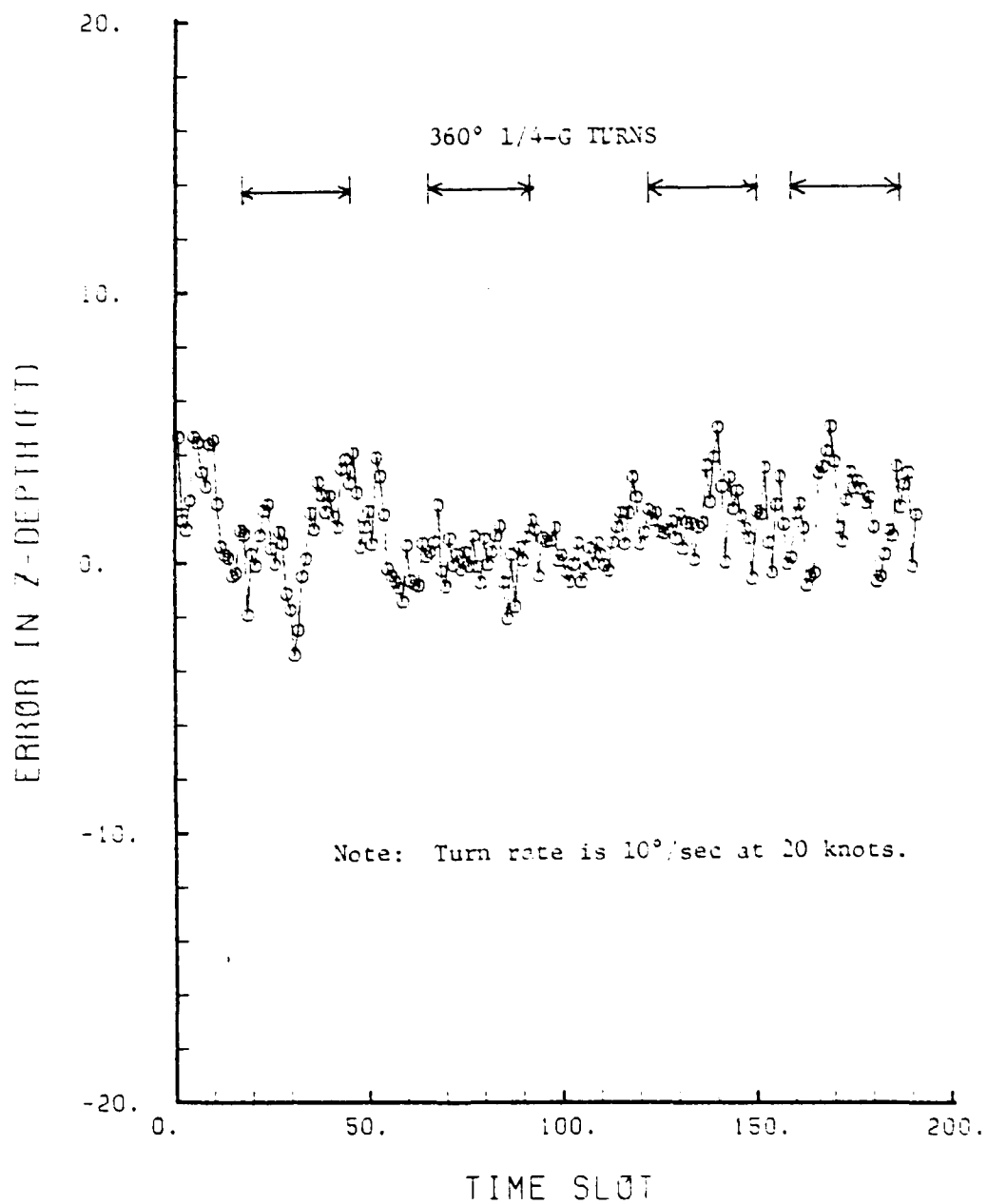


Figure 24. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

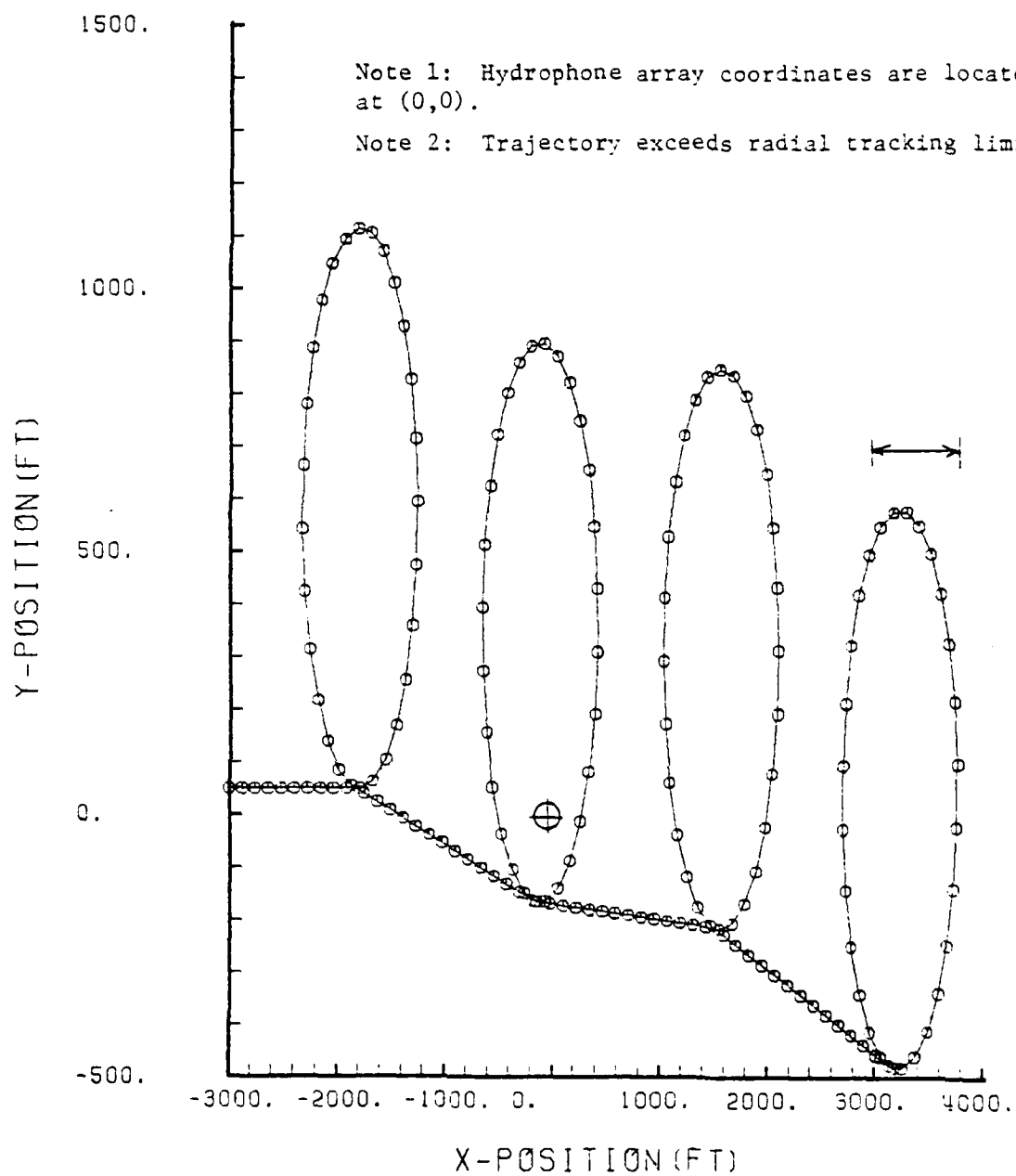


Figure 25. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

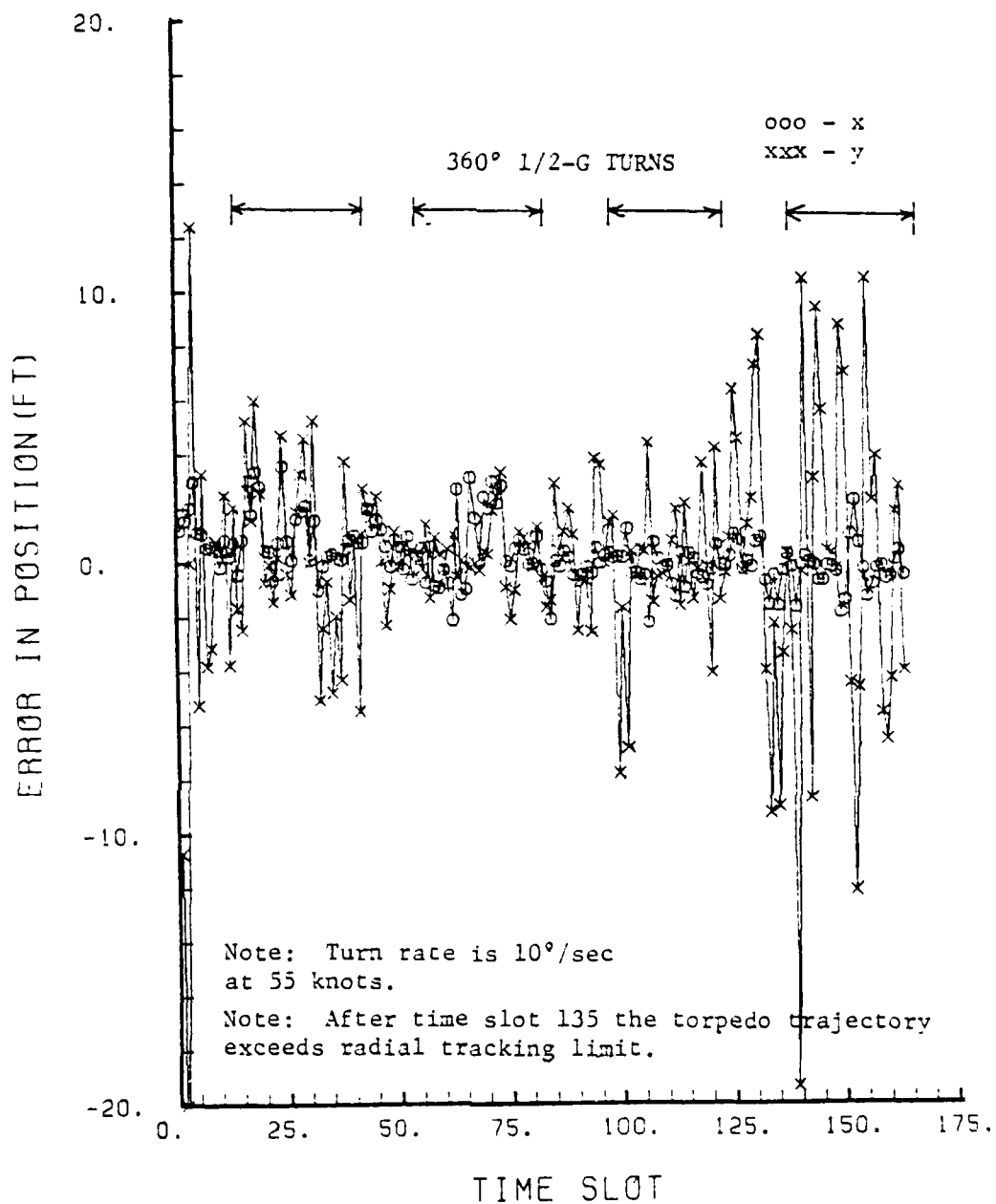


Figure 26. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

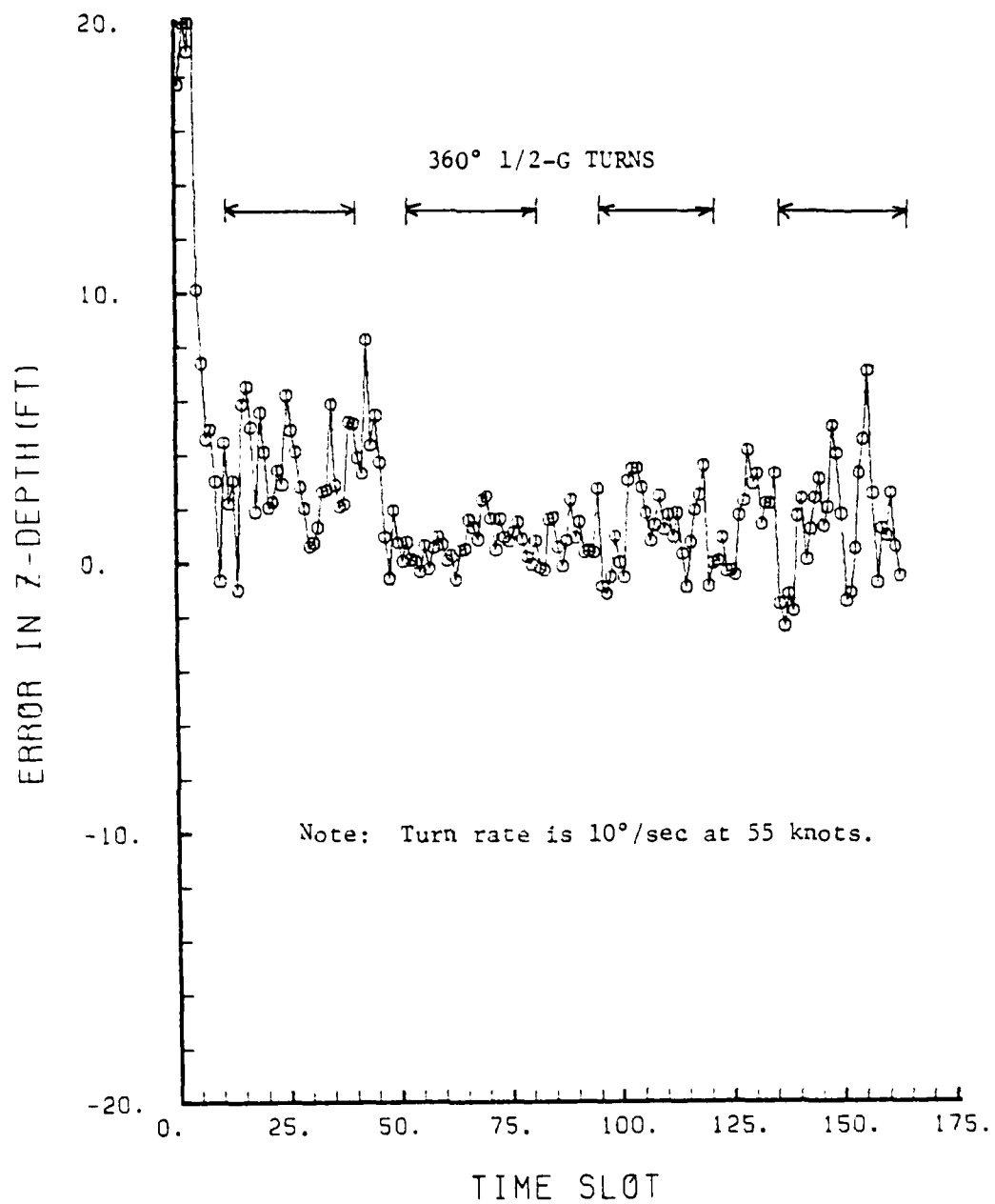


Figure 27. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

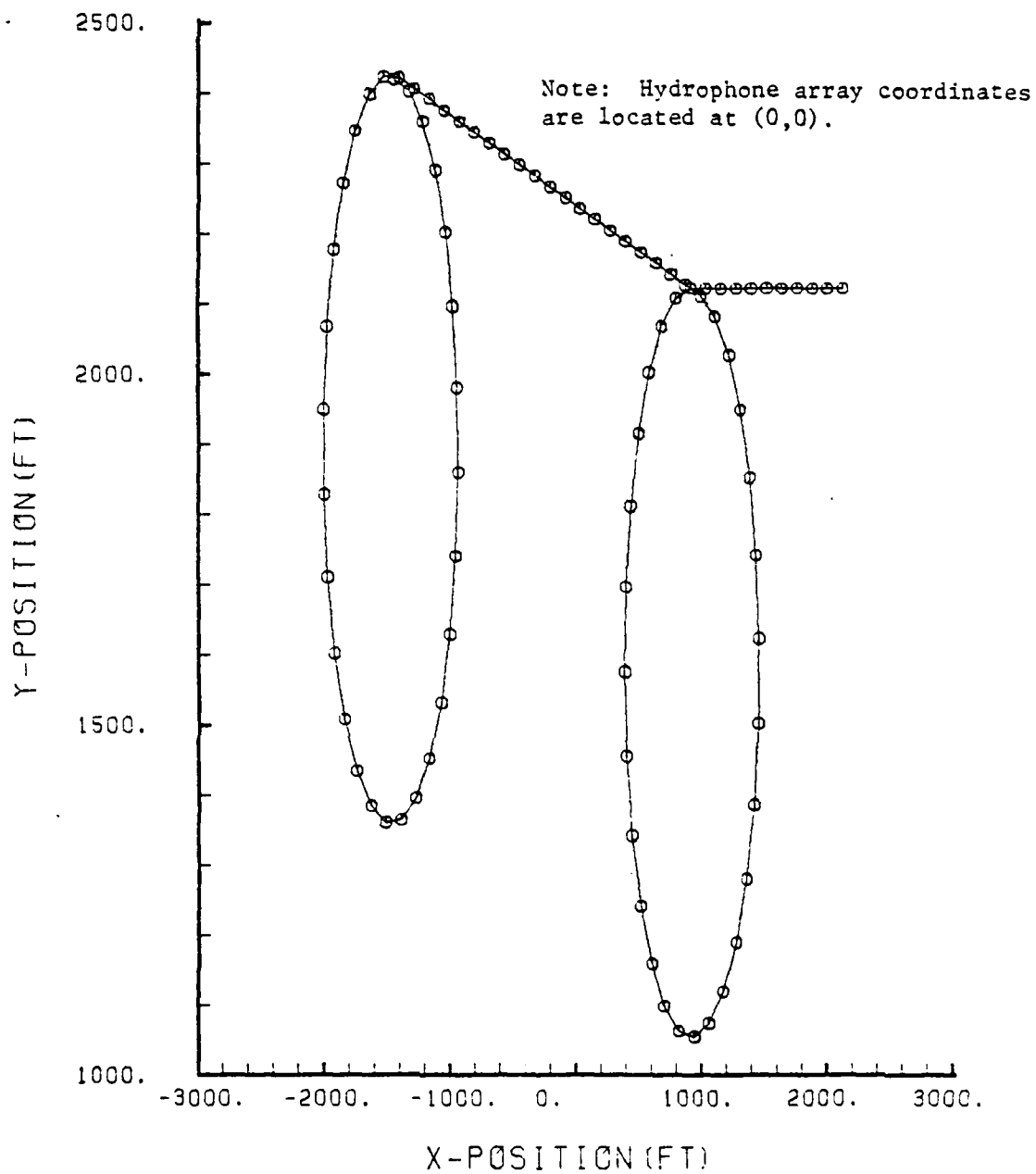


Figure 28. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

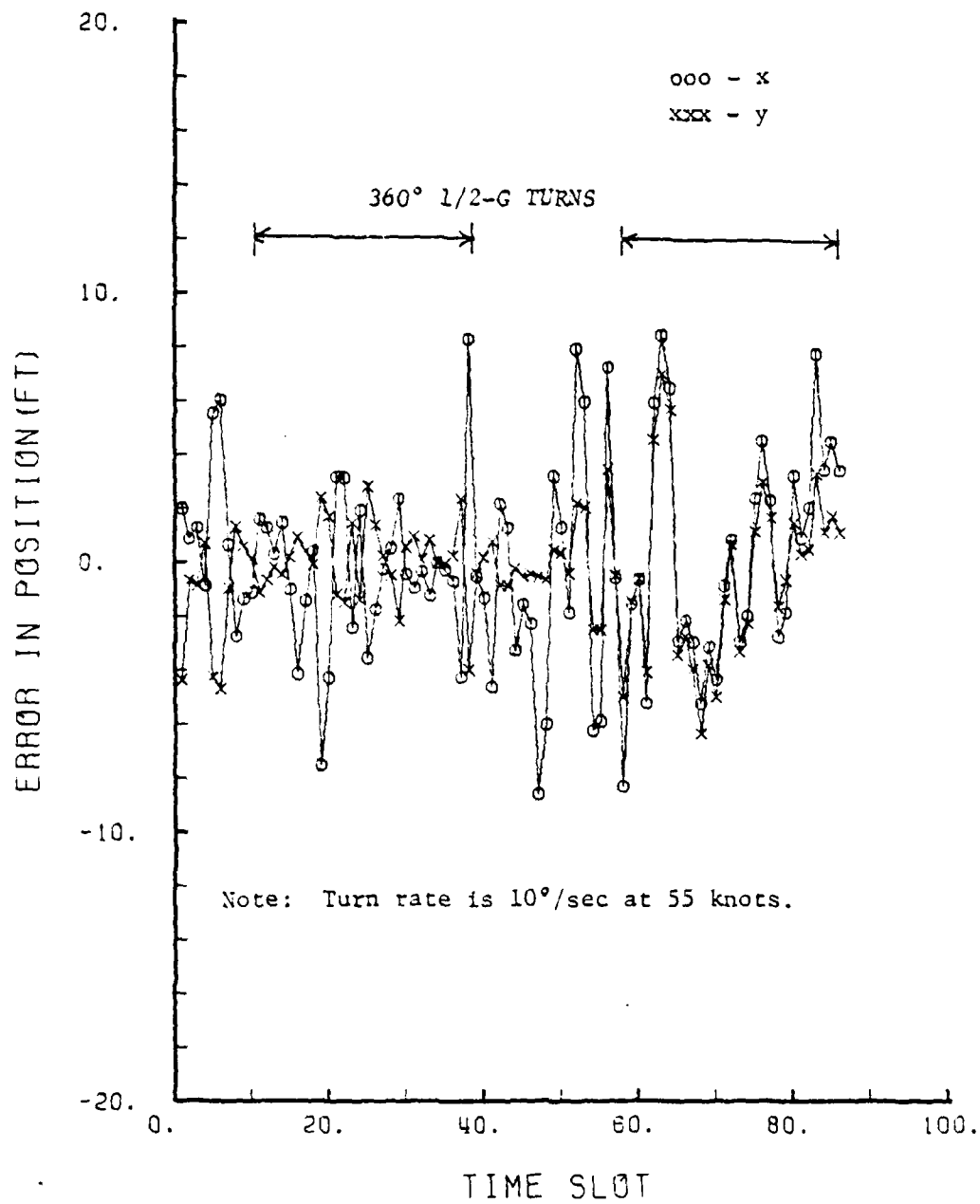


Figure 29. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

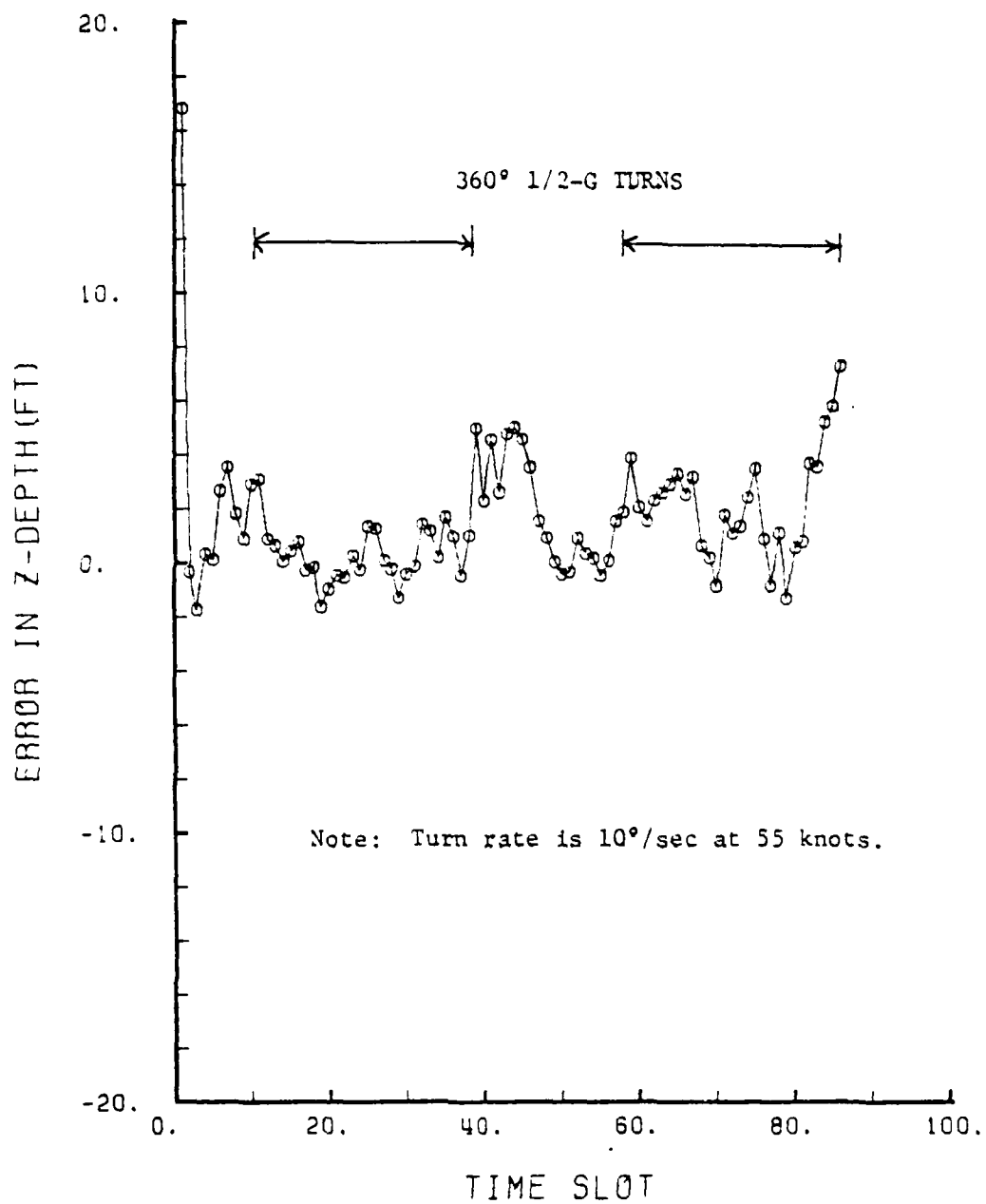


Figure 30. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

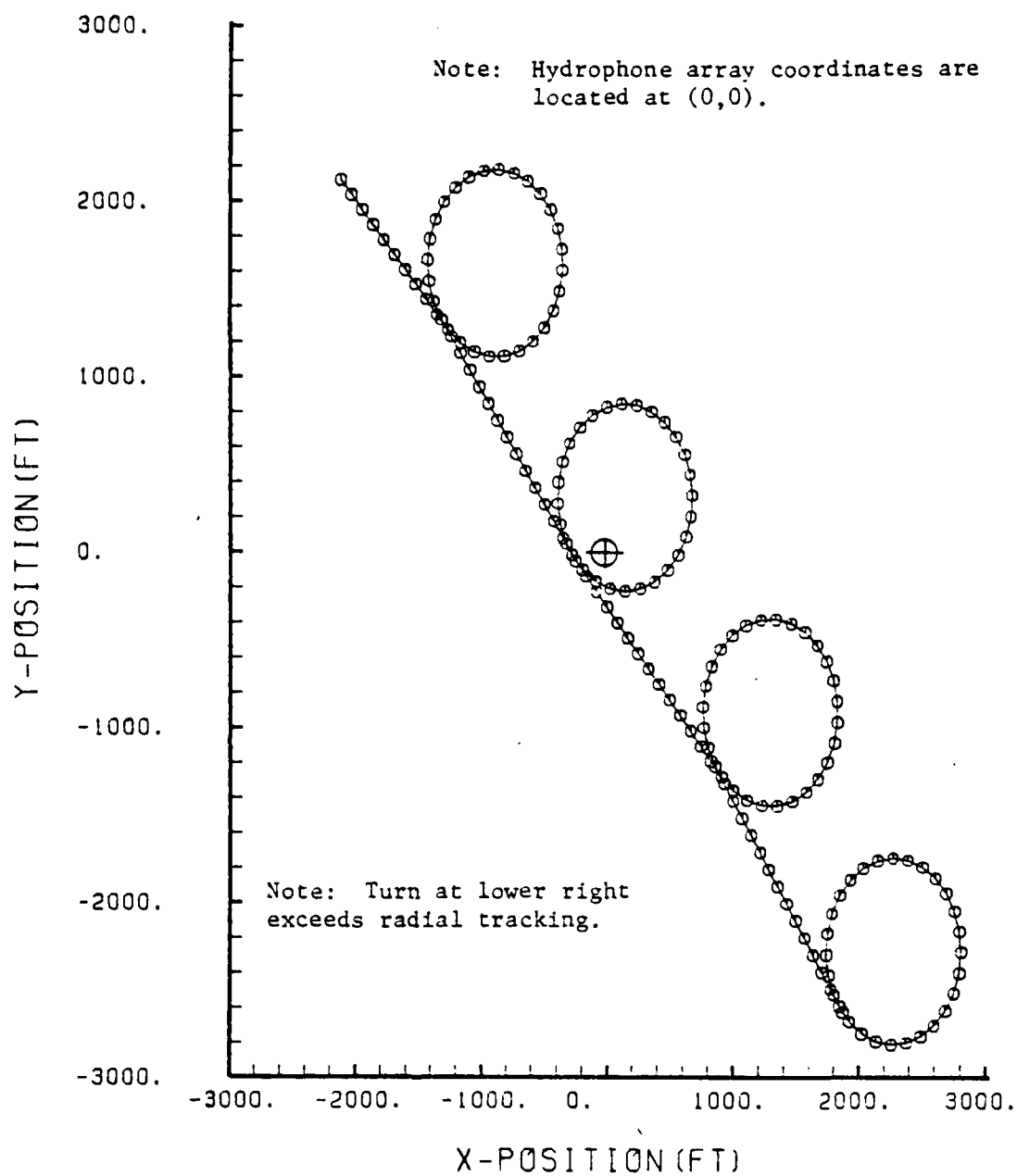


Figure 31. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

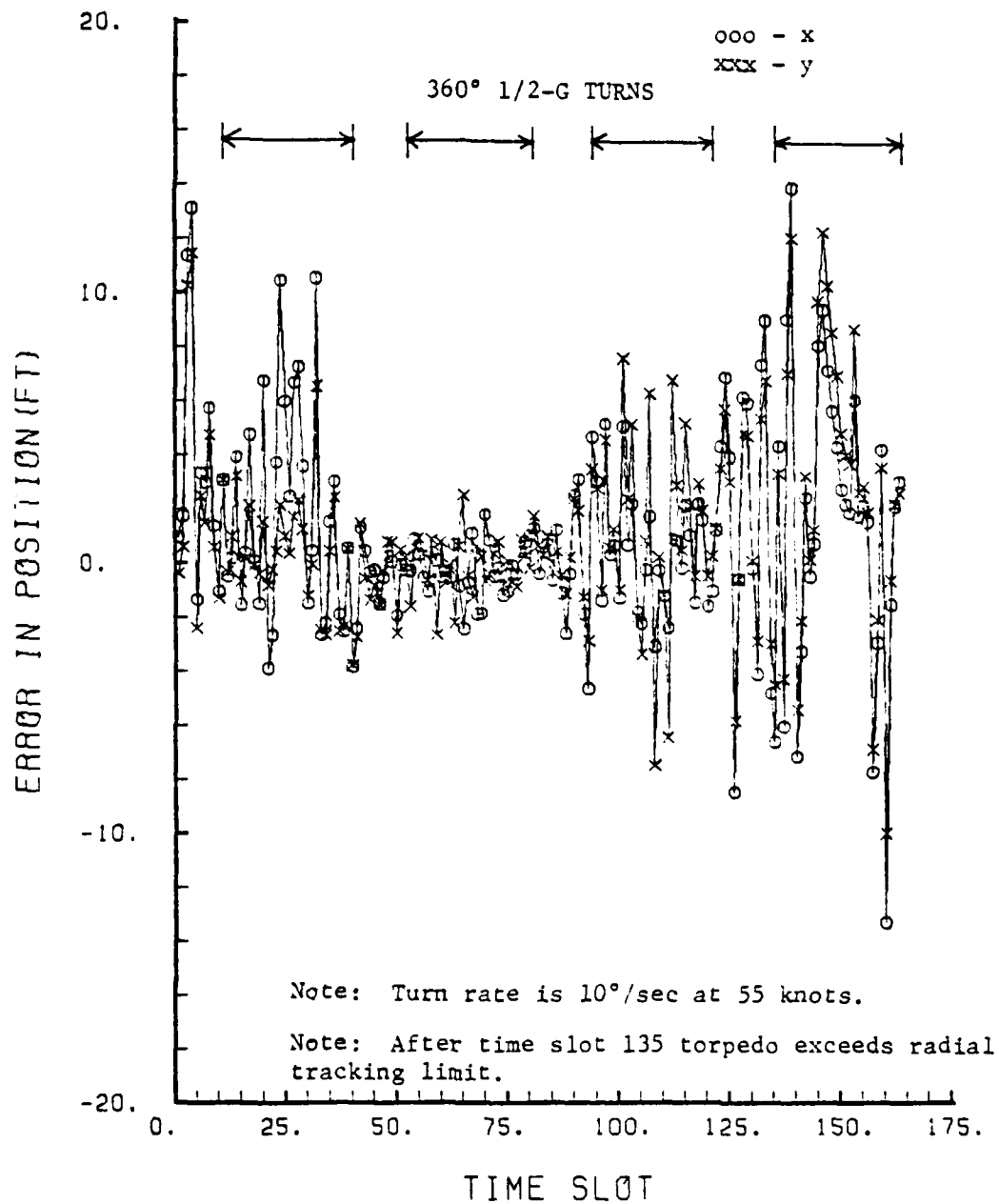


Figure 32. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

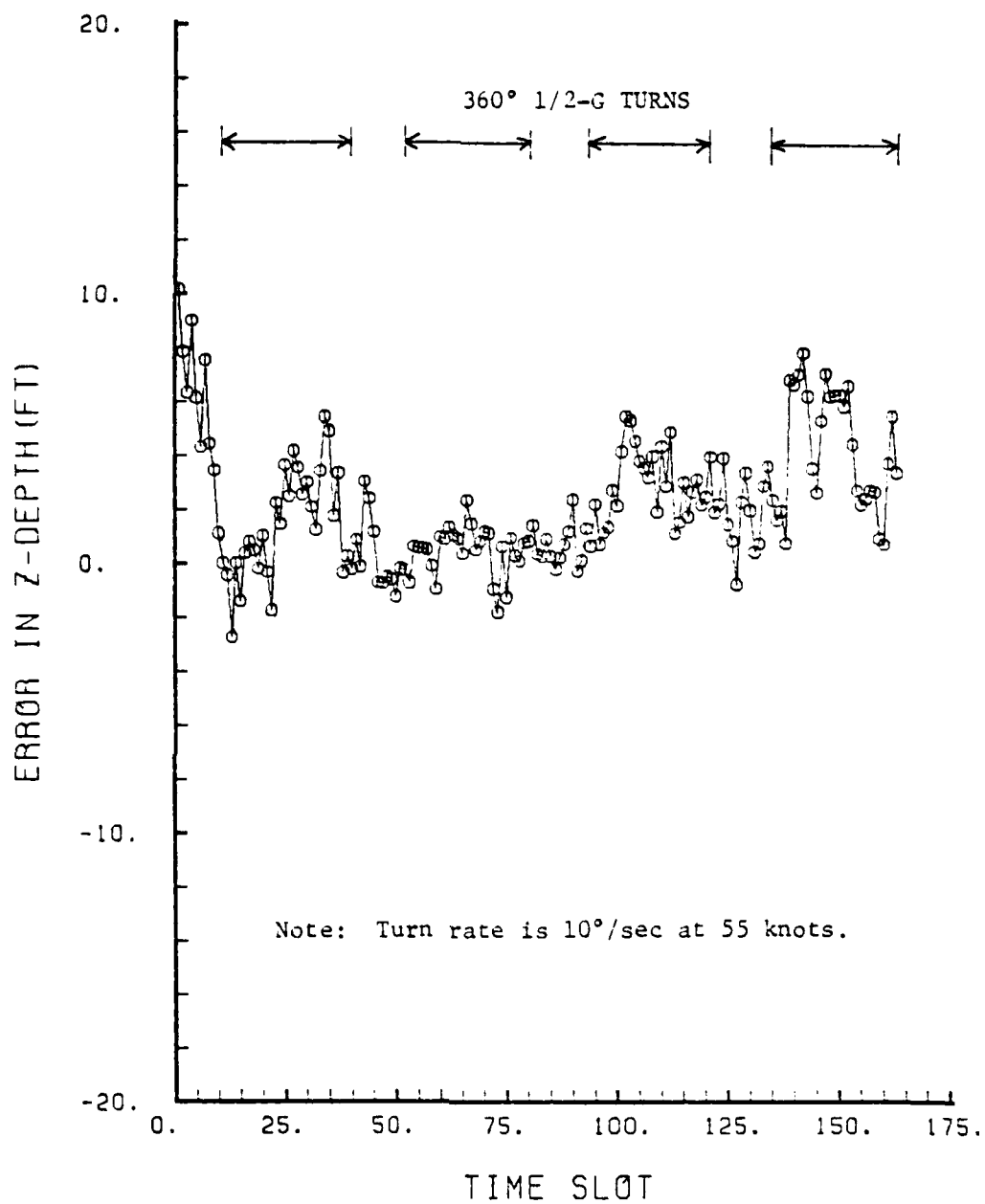


Figure 33. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

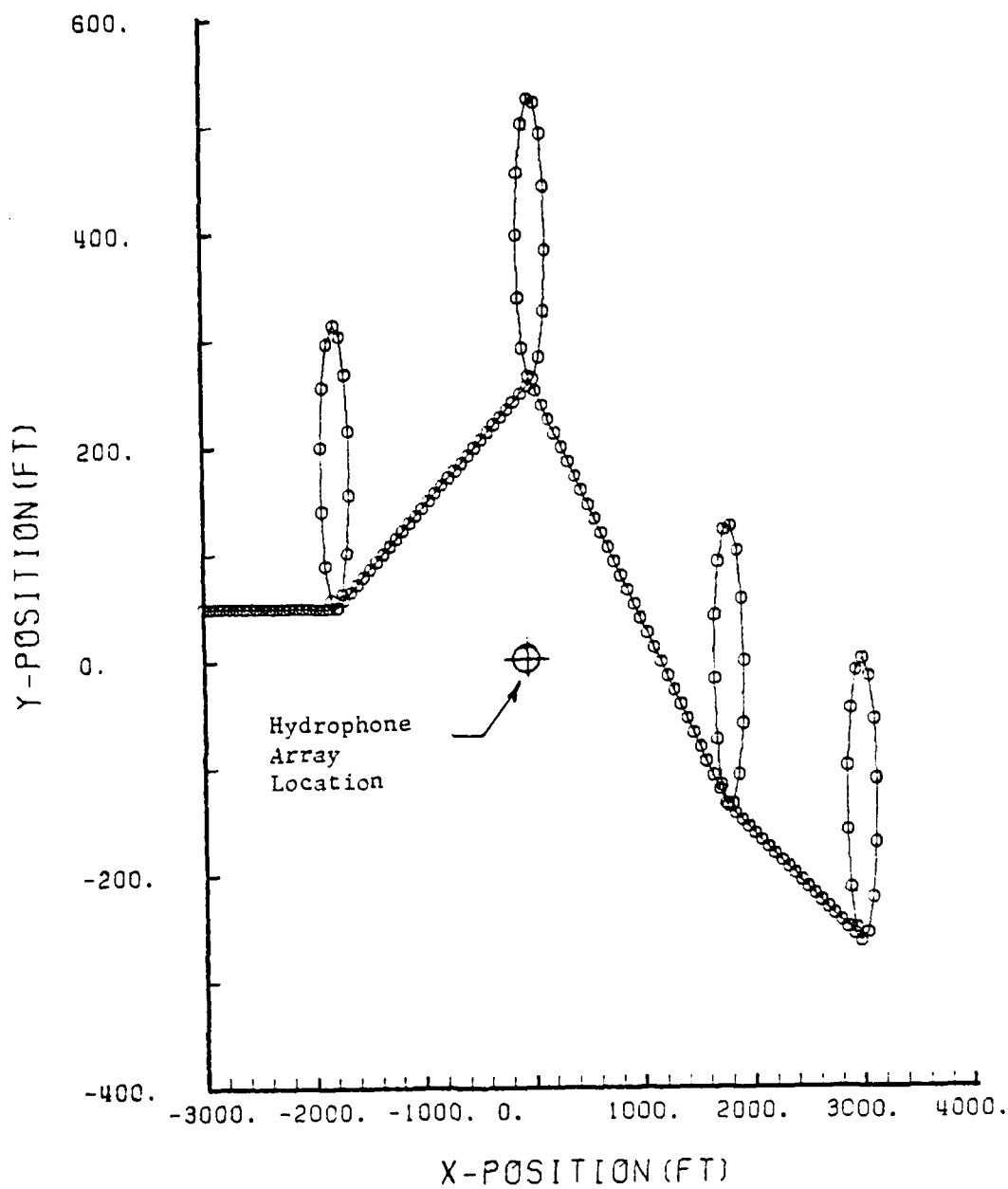


Figure 34. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

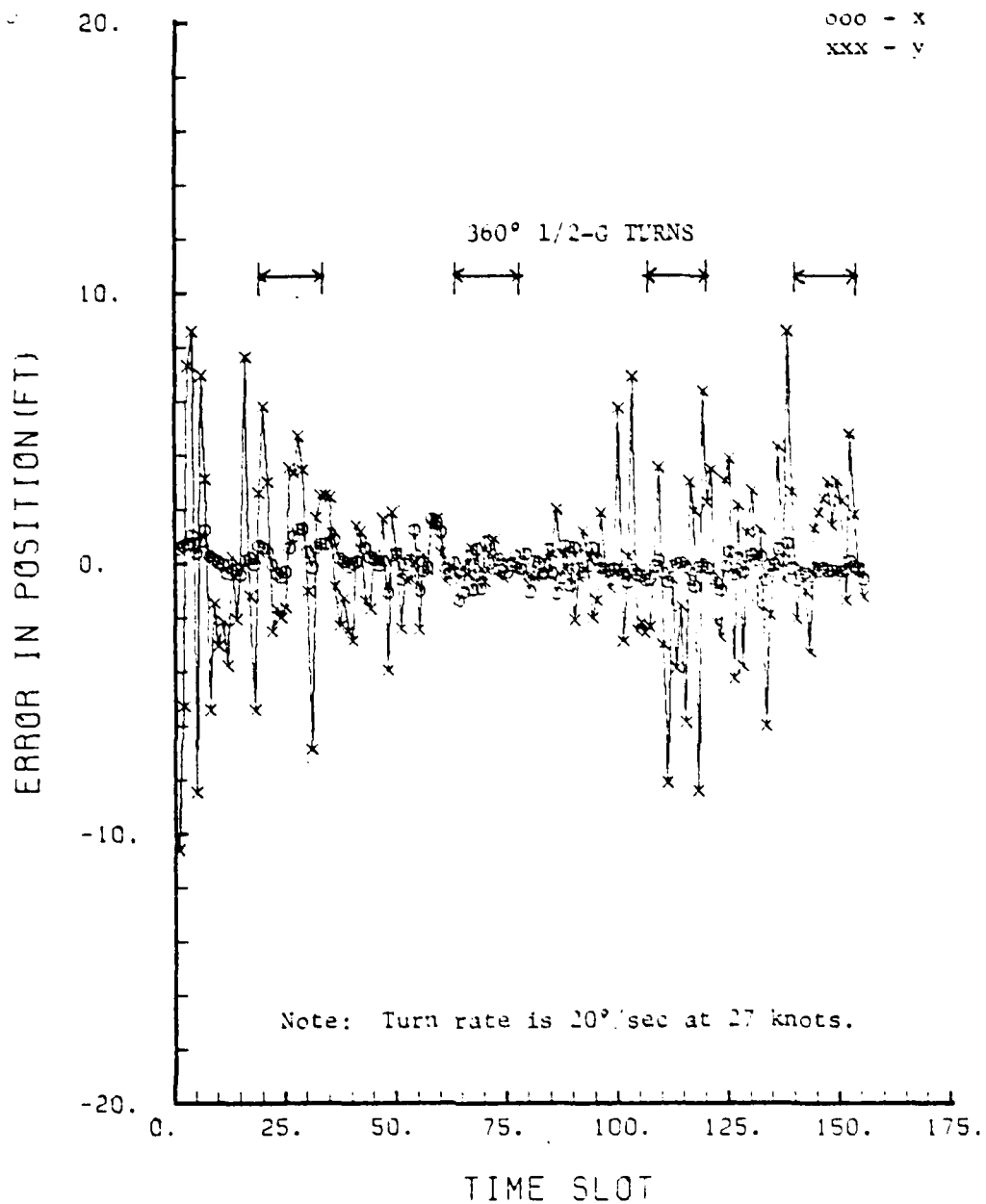


Figure 35. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

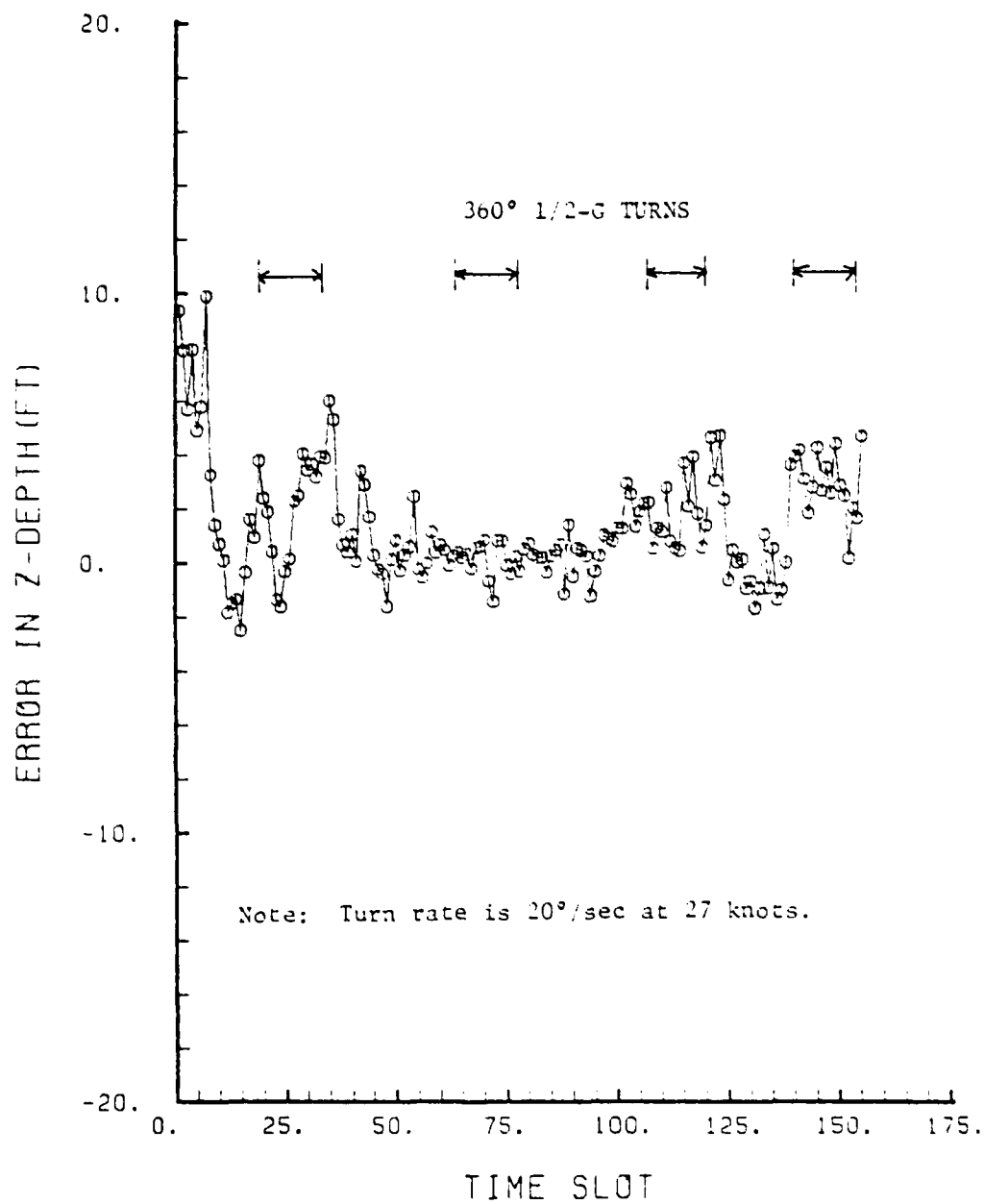


Figure 36. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

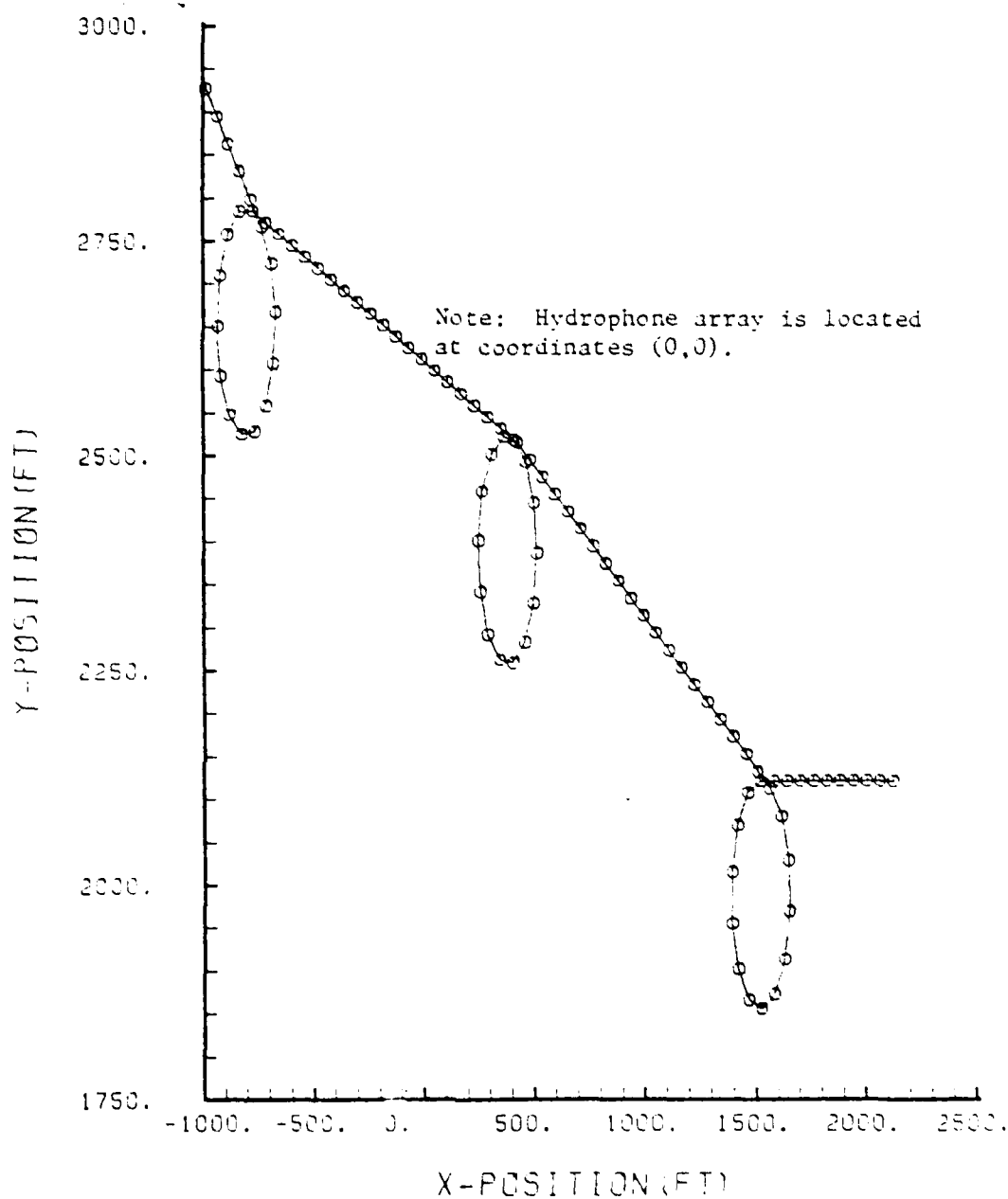


Figure 37. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

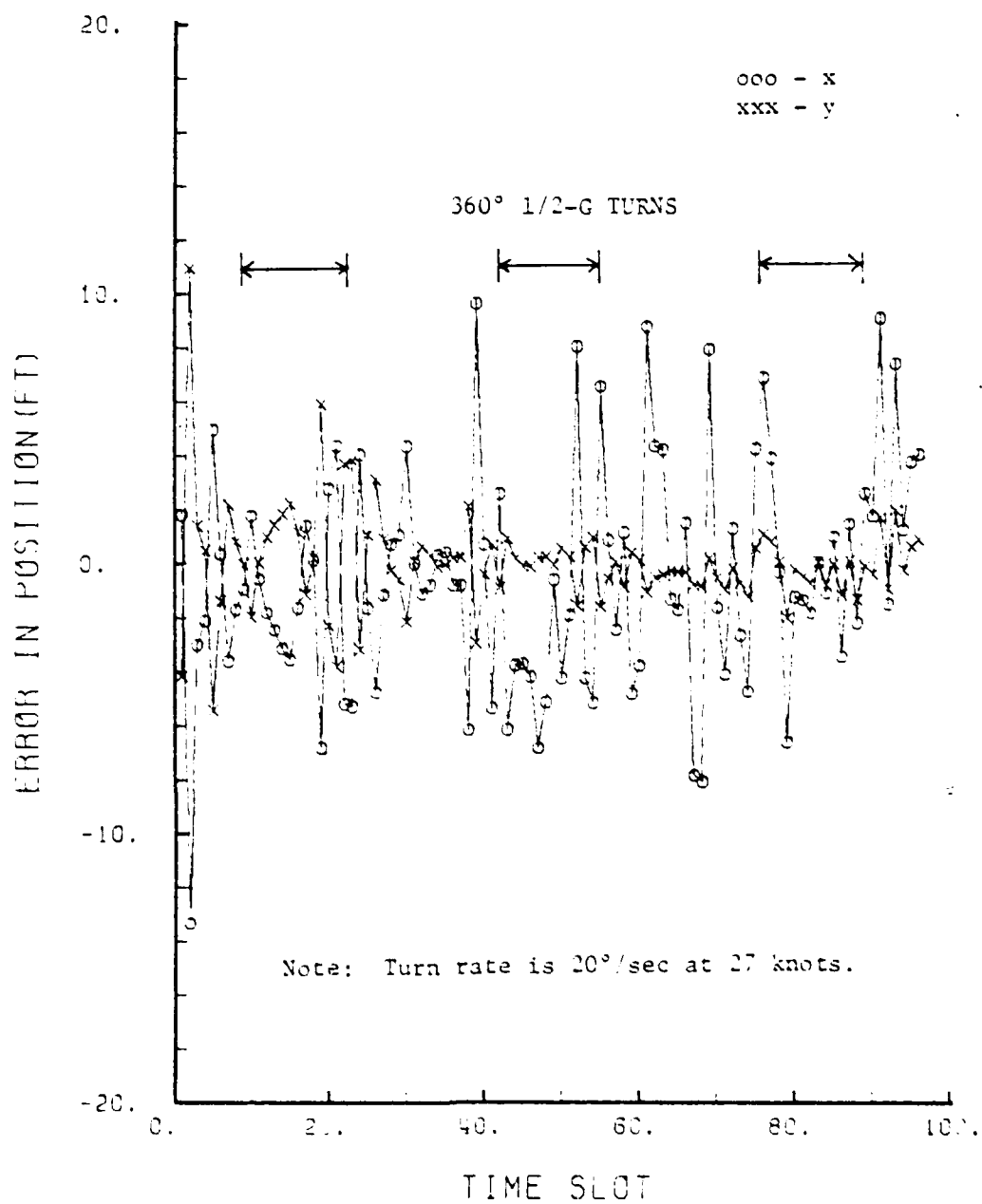


Figure 38. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

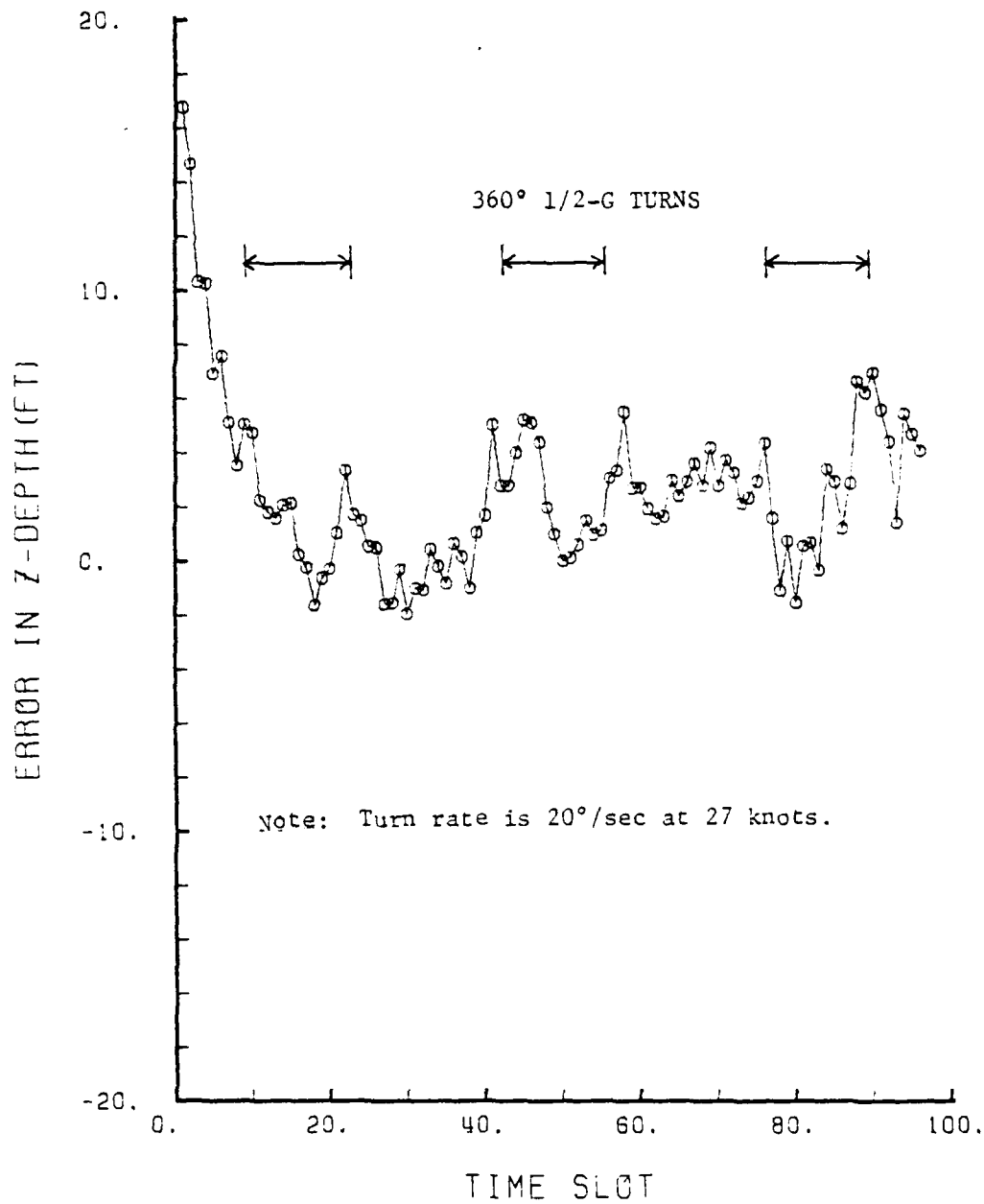


Figure 39. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

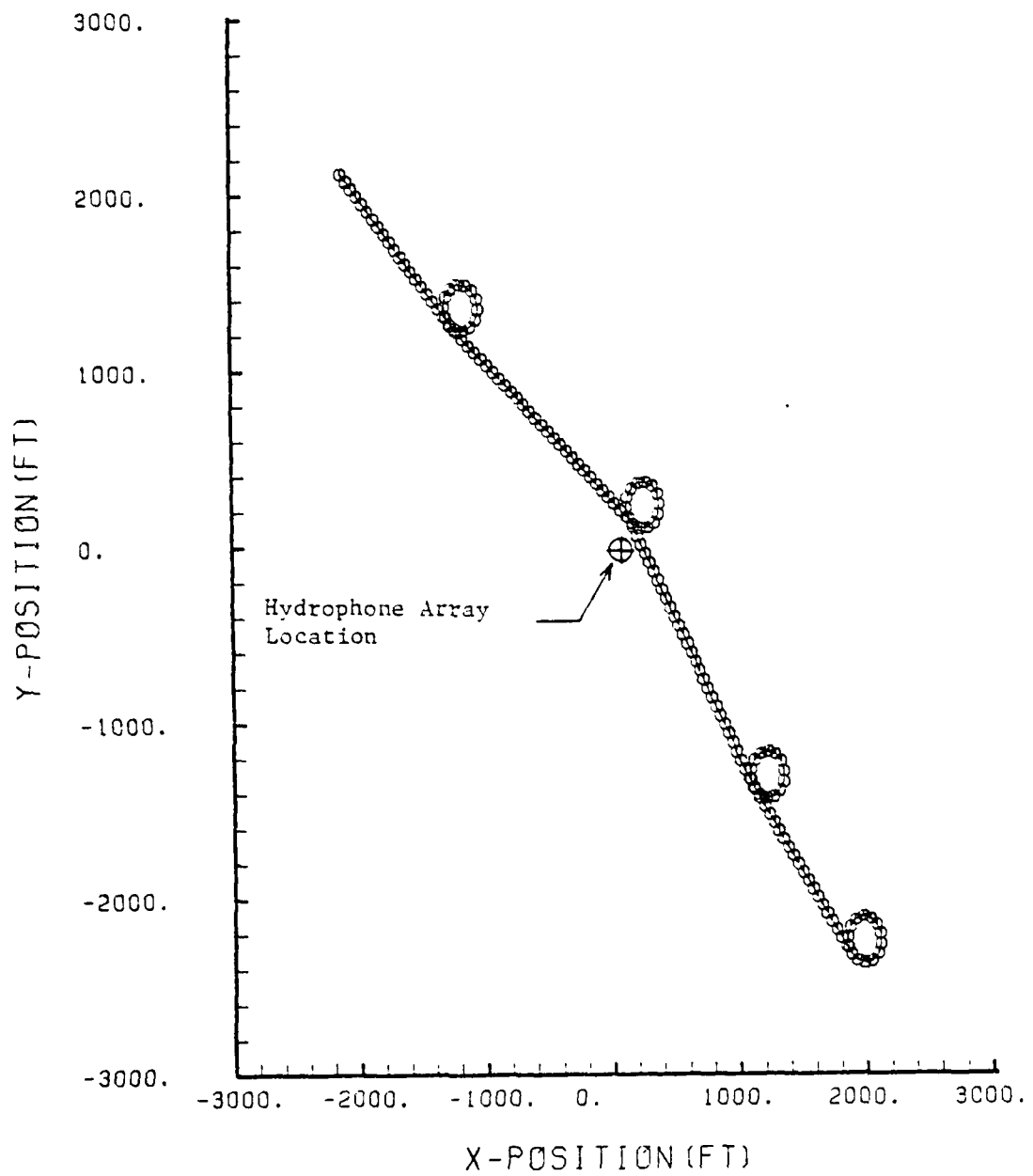


Figure 40. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

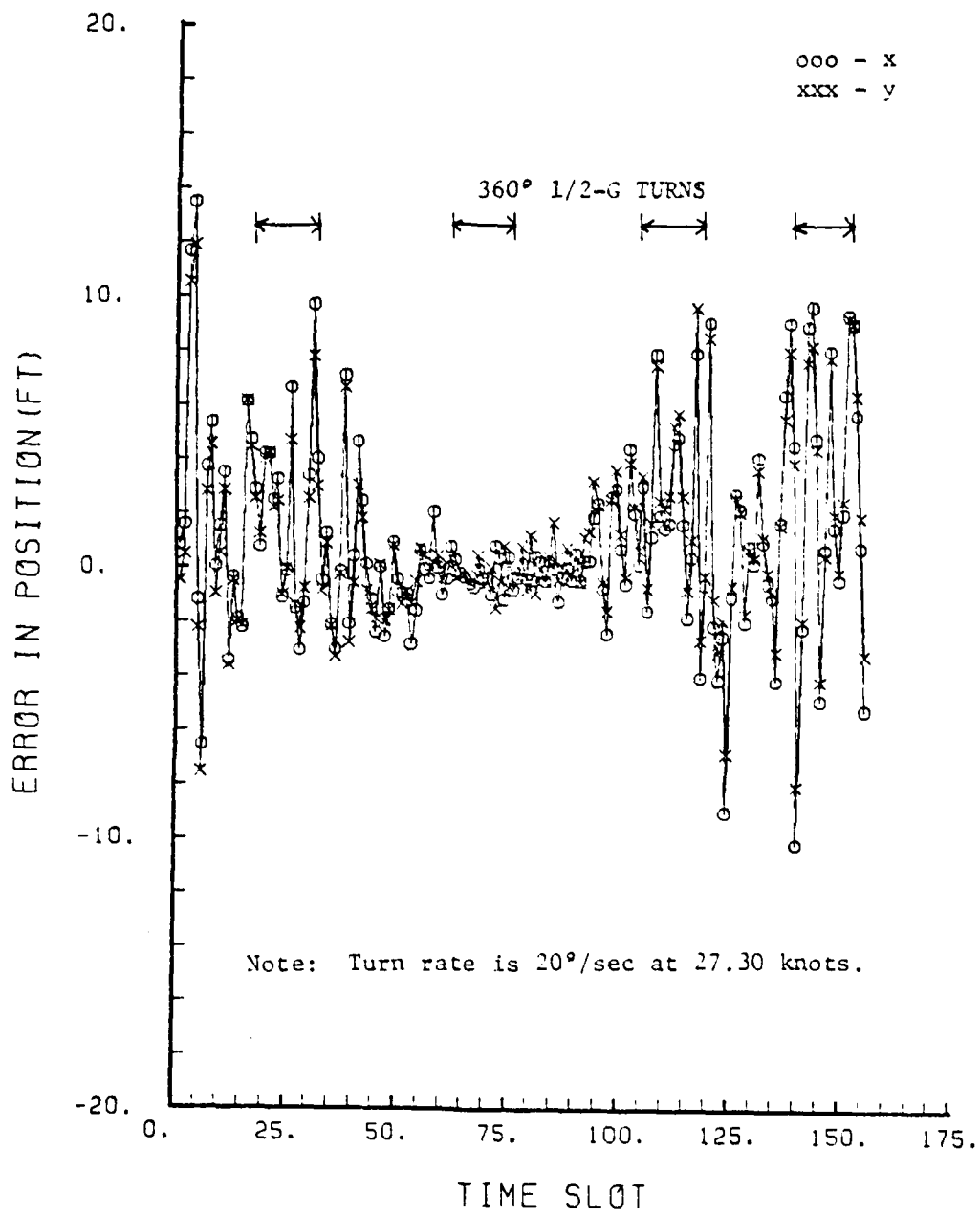


Figure 41. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

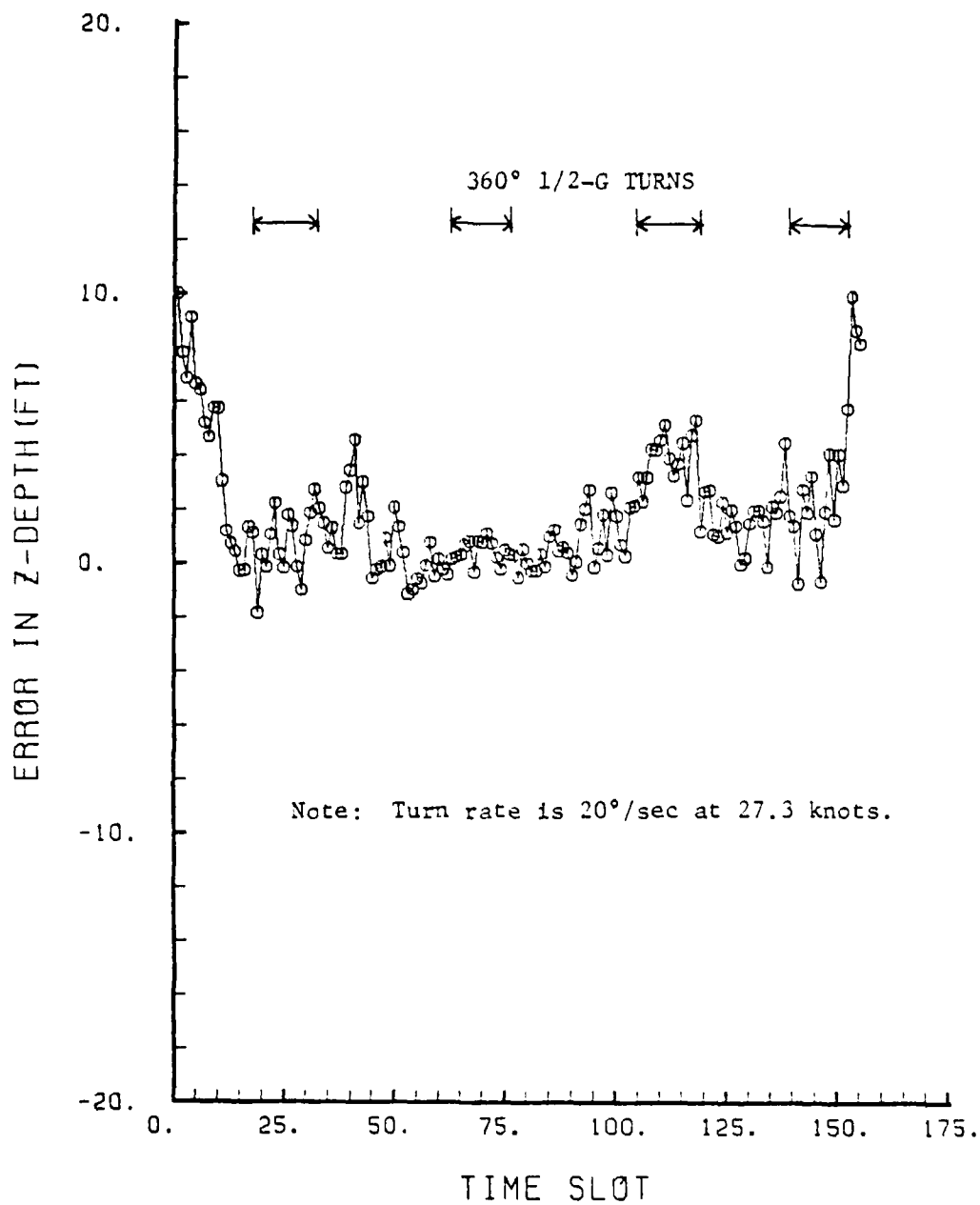


Figure 42. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

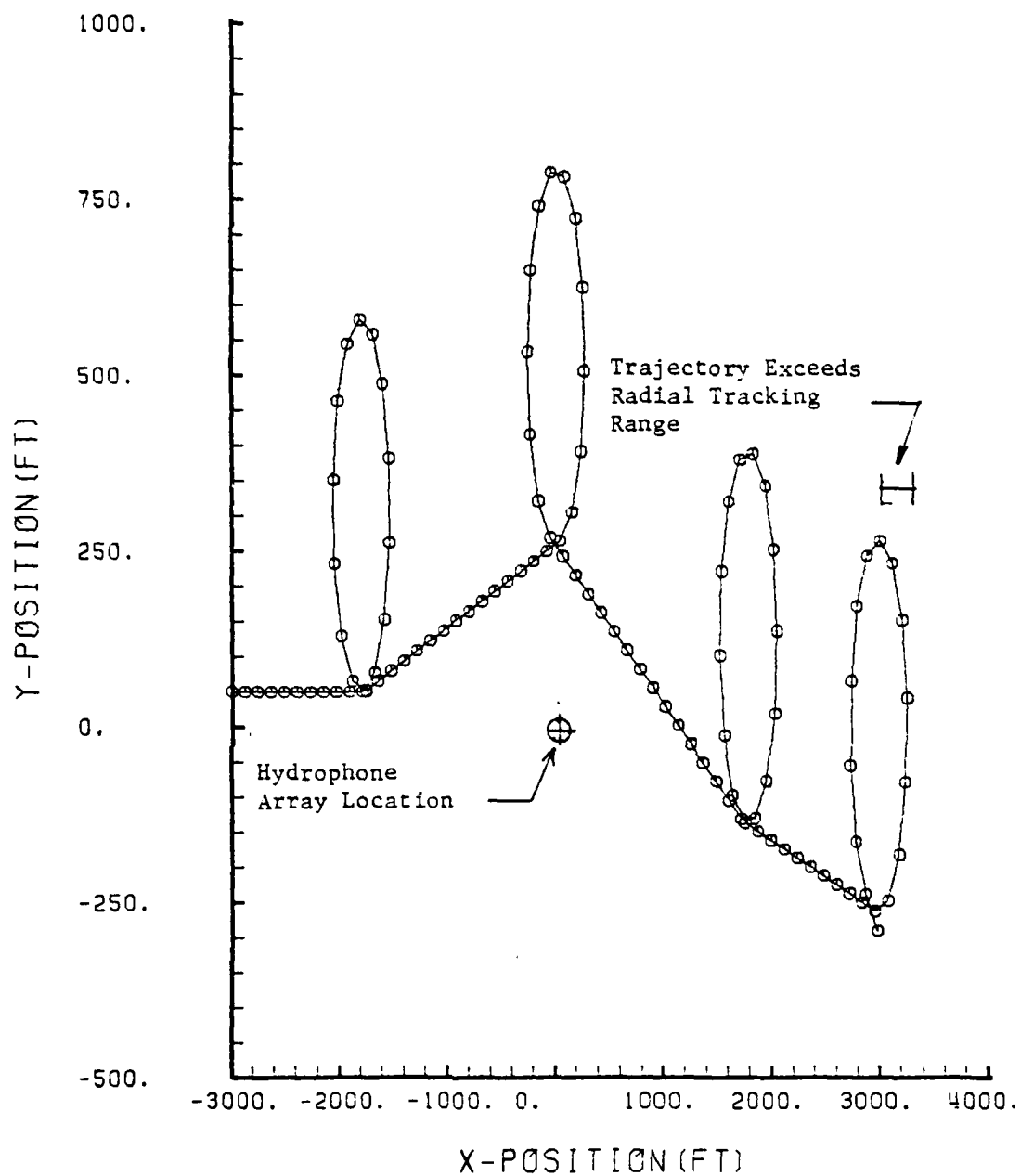


Figure 43. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

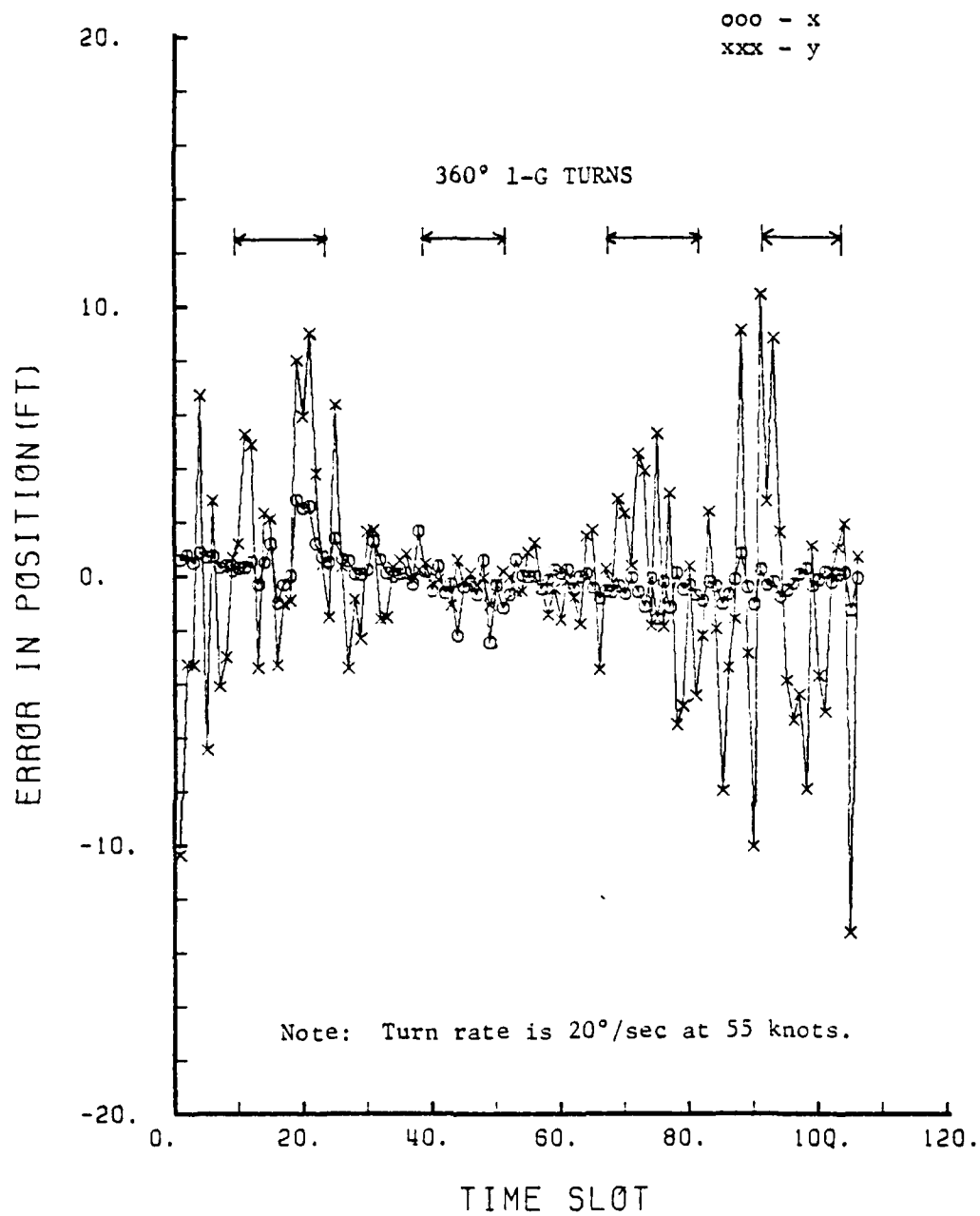


Figure 44. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

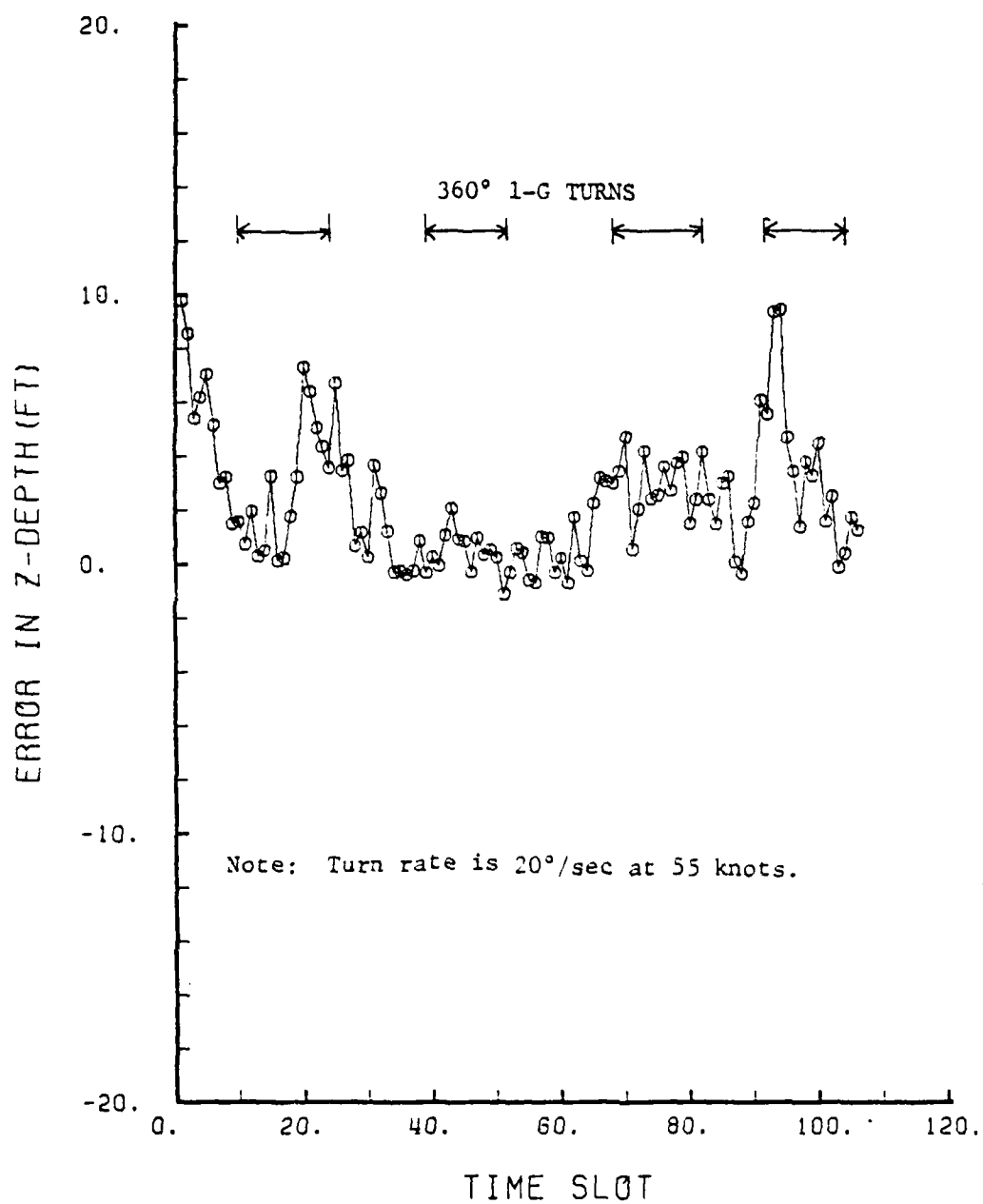


Figure 45. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

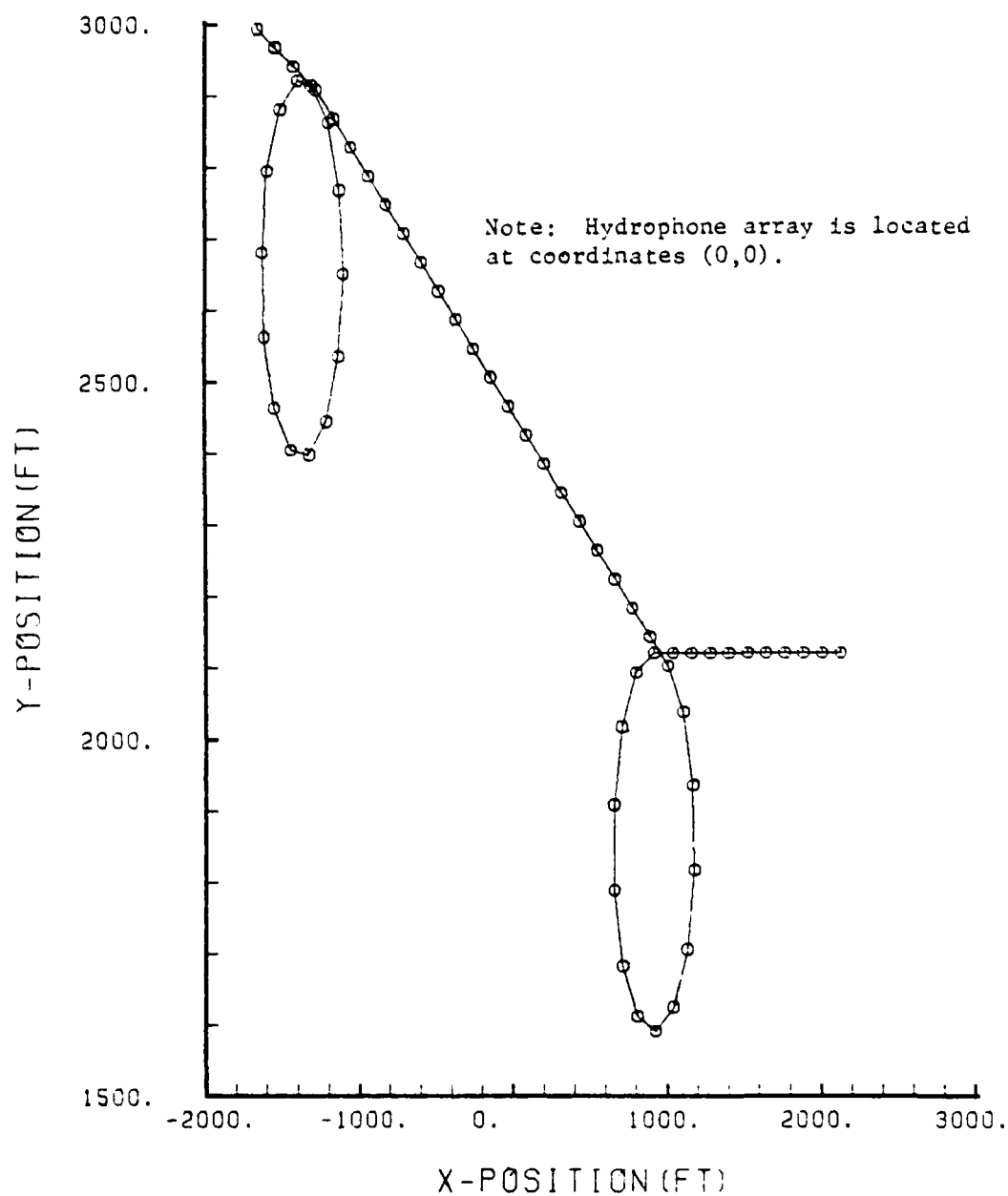


Figure 46. True Trajectory of the Torpedo During a Maneuvering Run in the Area of A Single Array

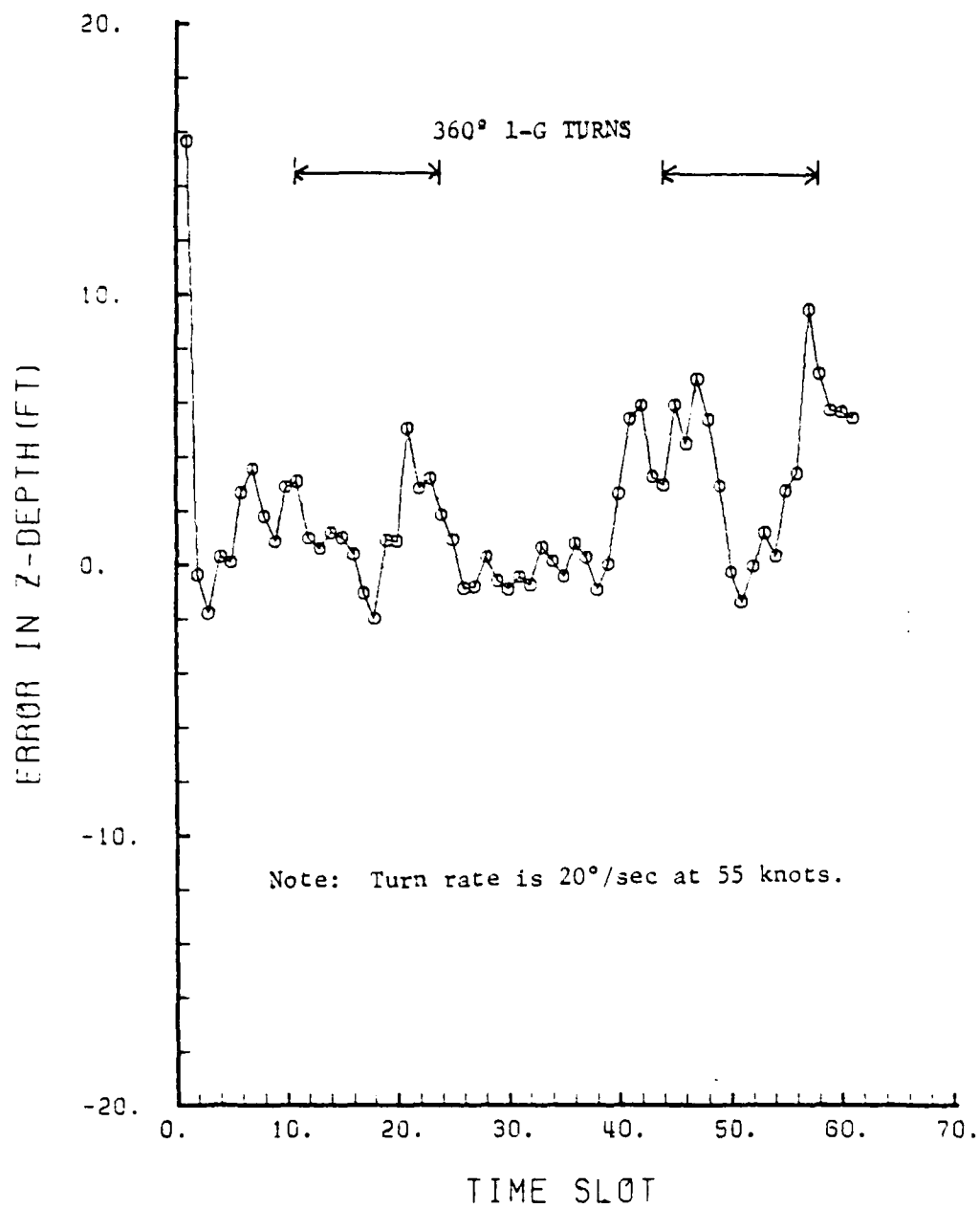


Figure 47. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

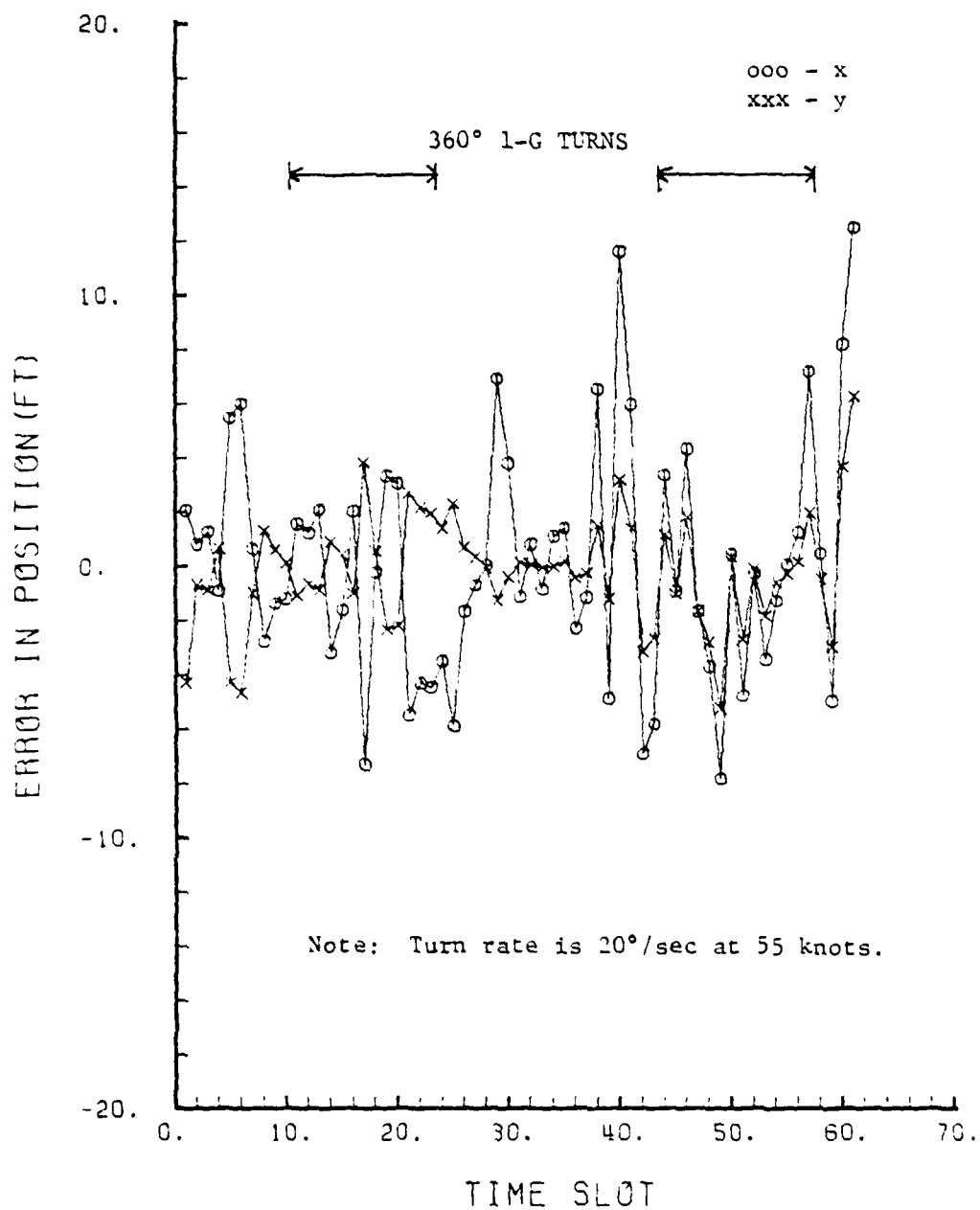


Figure 48. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

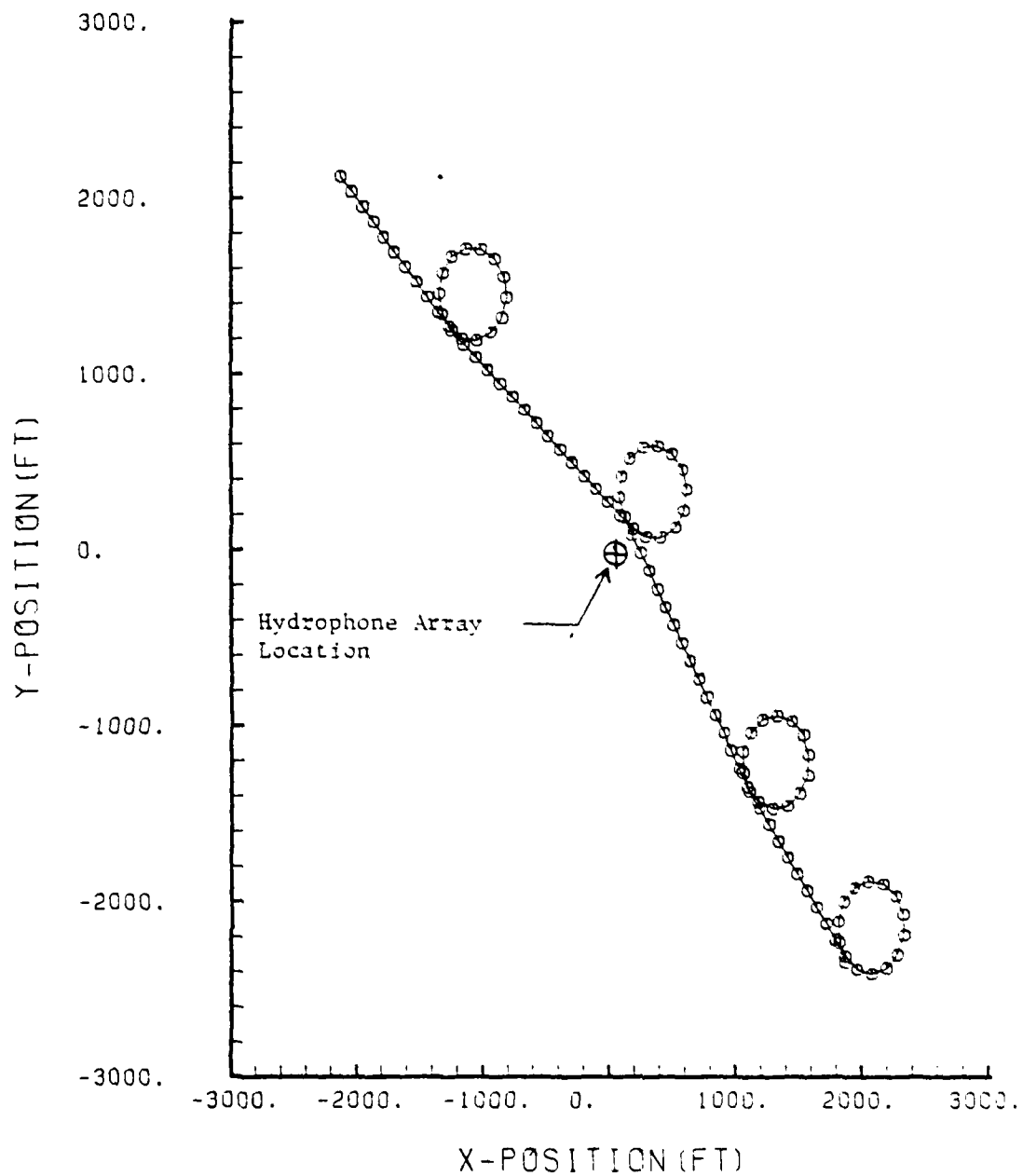


Figure 49. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

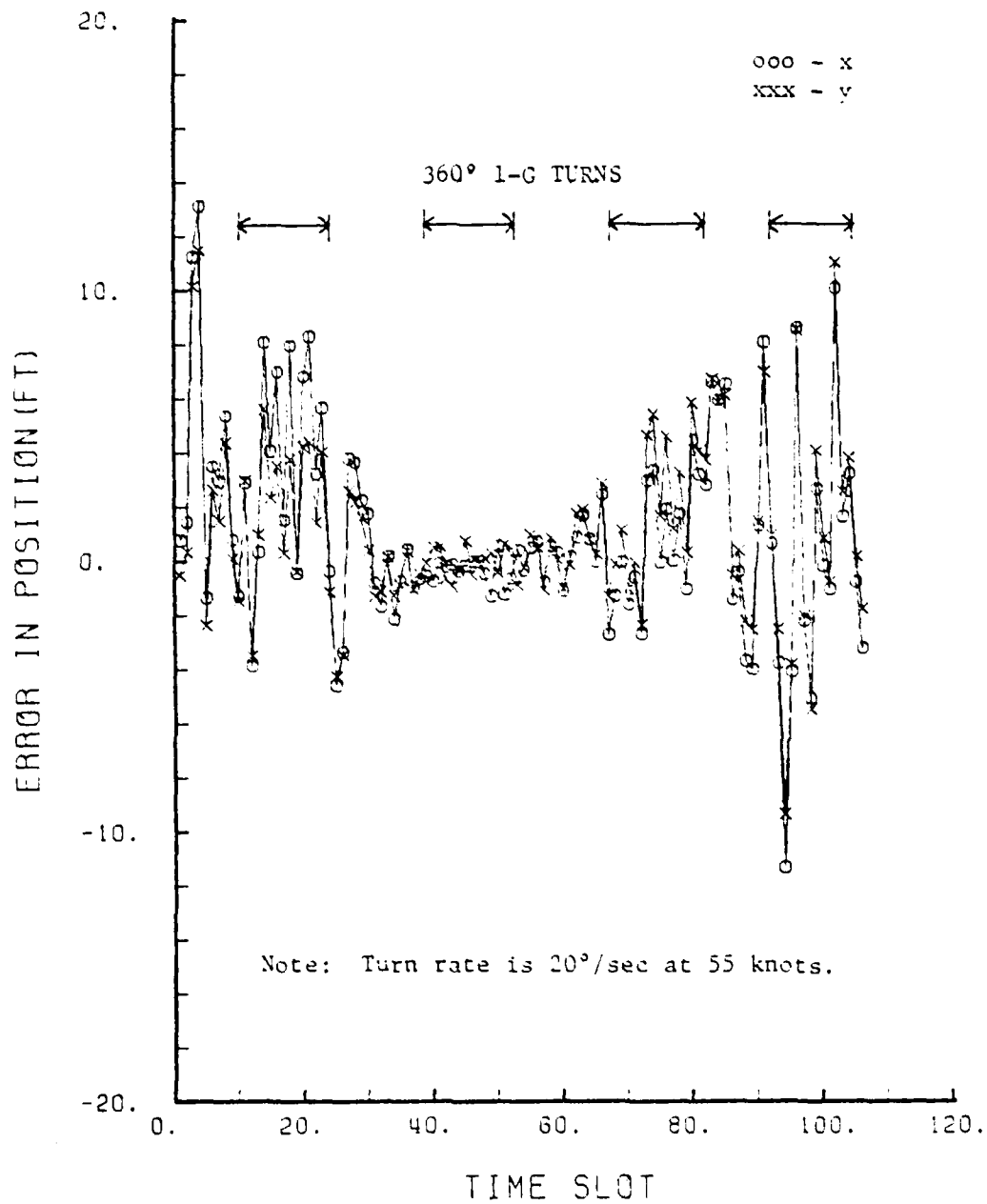


Figure 50. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

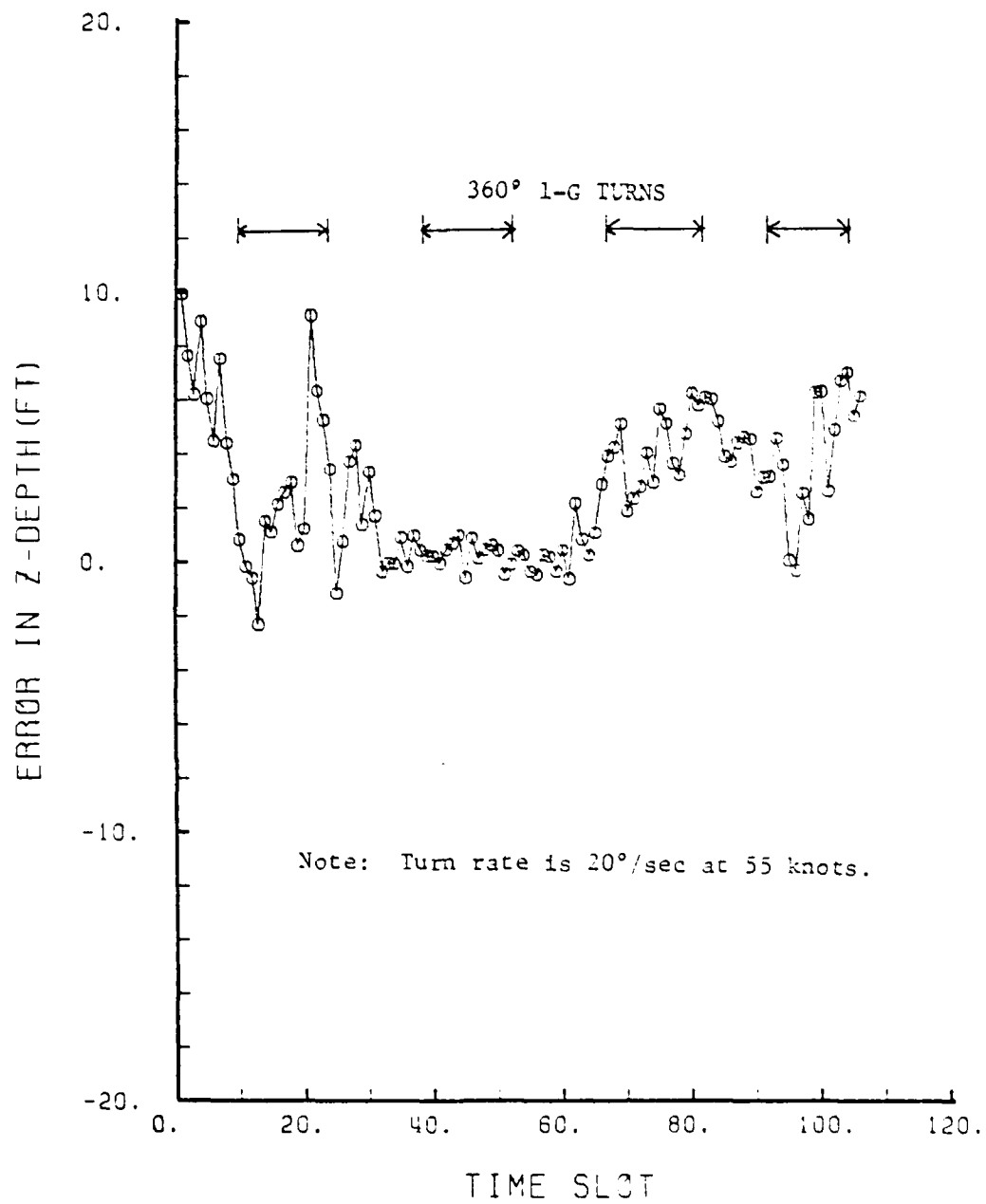


Figure 51. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

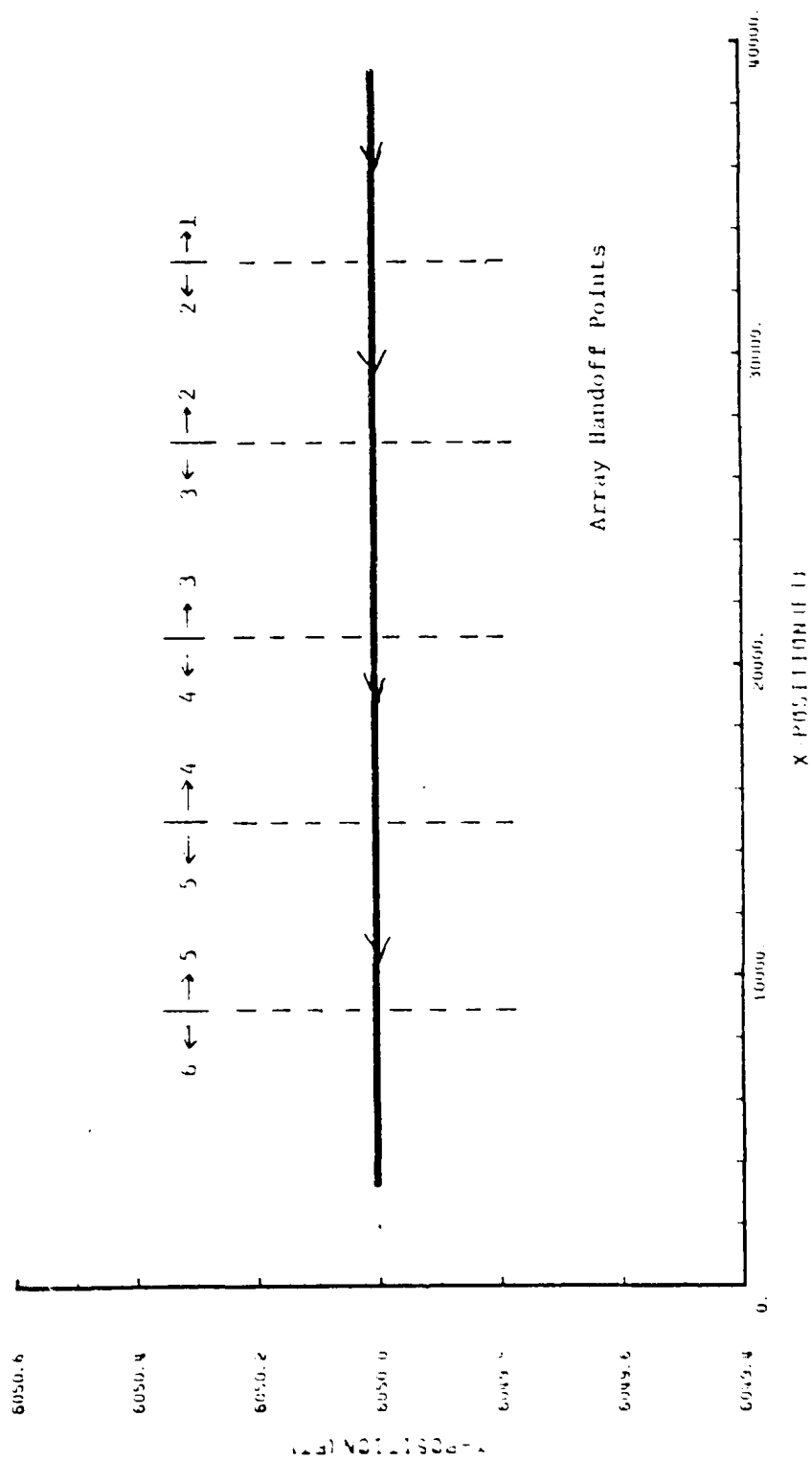


Figure 52. True Trajectory of the Torpedo during a Straight Run Through Multiple Arrays

AD-A092 303

NAVAL POSTGRADUATE SCHOOL MONTEREY CA
AN APPLICATION OF KALMAN FILTERING TO TORPEDO TRACKING.(U)
SEP 80 P A O'BRIEN

F/G 17/1

UNCLASSIFIED

NL

2 of 2

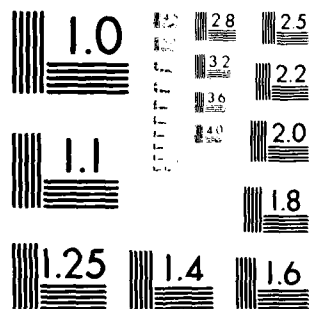
AL
ADDITION

END

DATE

181

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

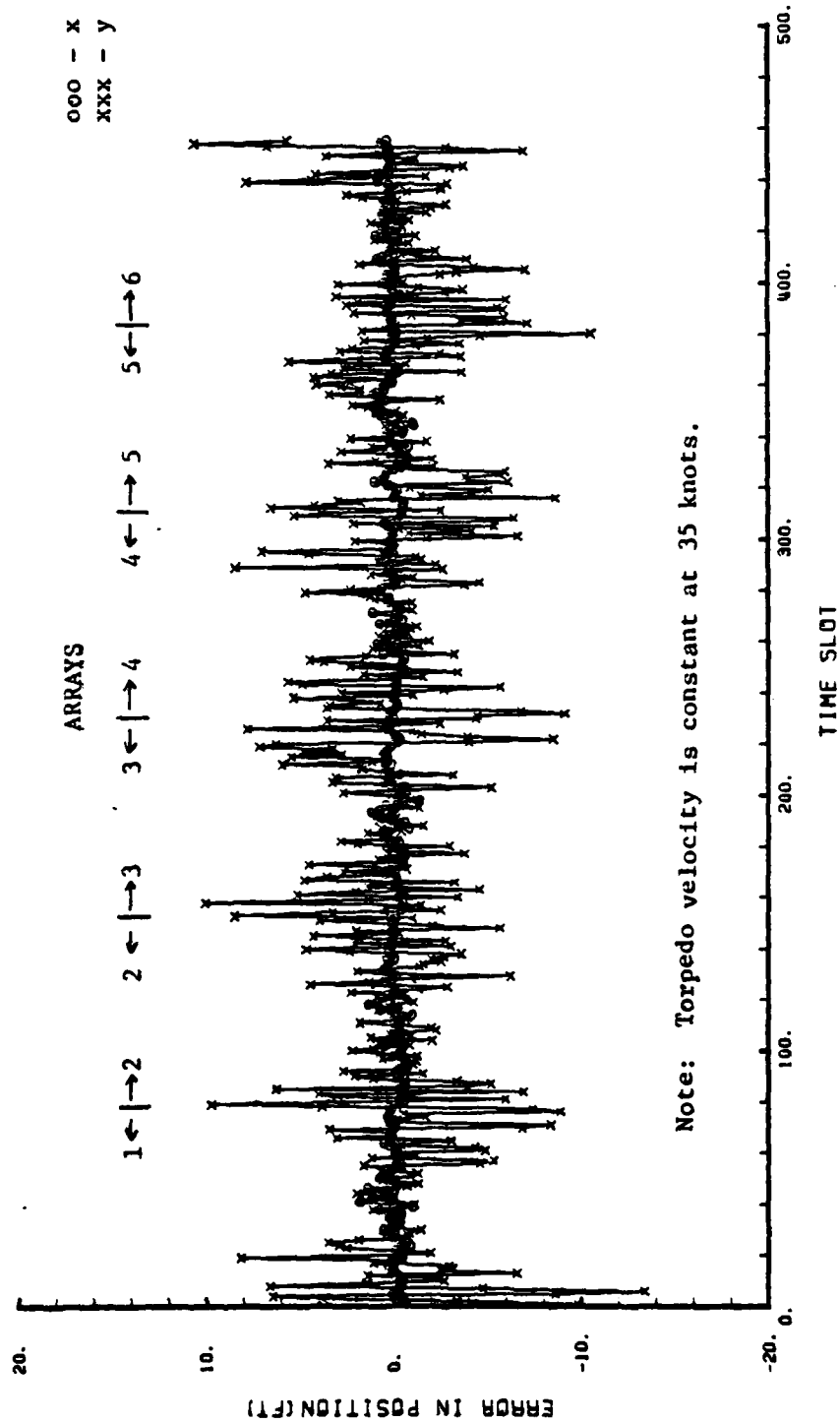


Figure 53. Error in Torpedo Position During a Straight Run Through Multiple Arrays

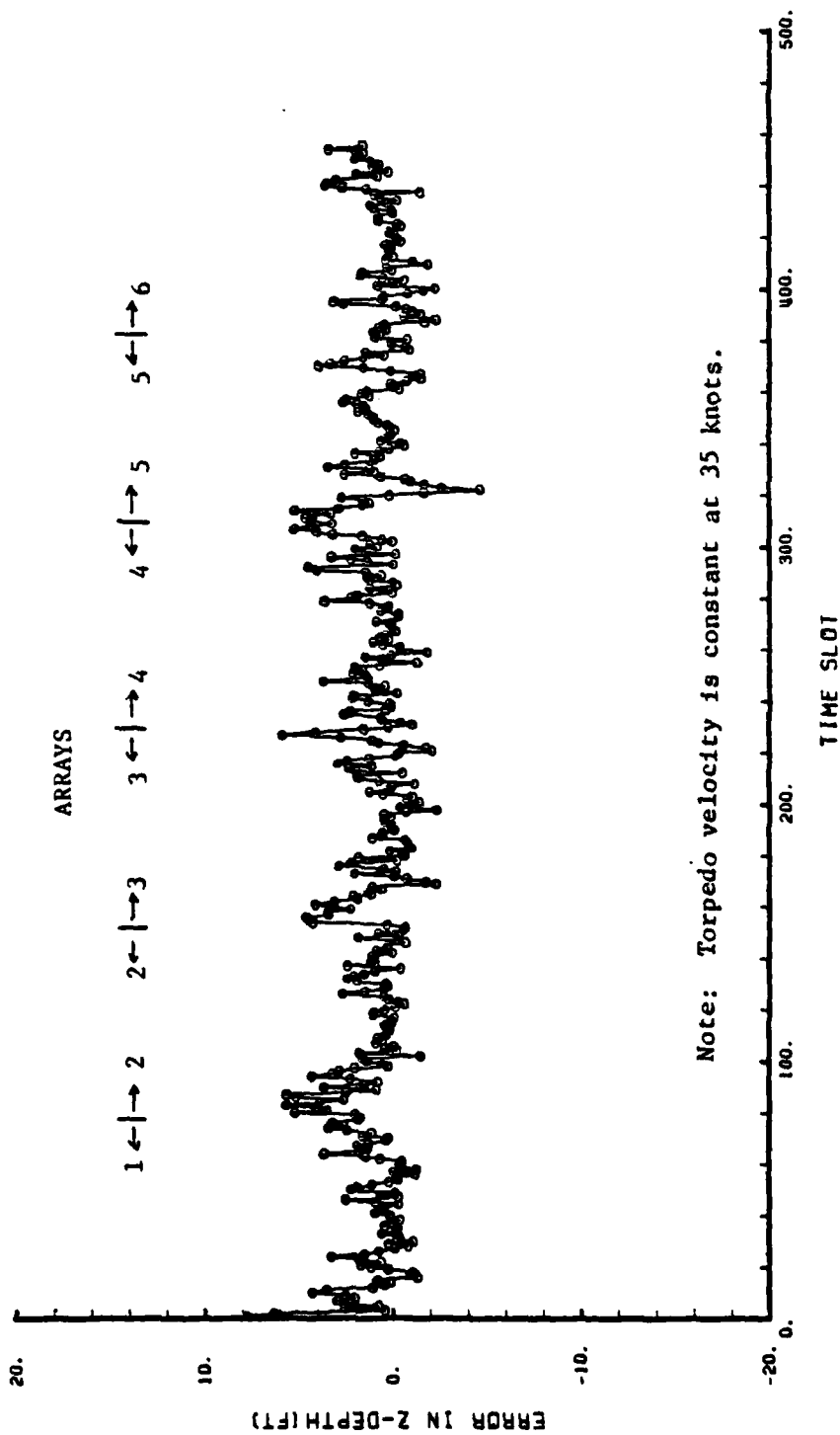


Figure 54. Error in Torpedo Depth During a Straight Run Through Multiple Arrays

APPENDIX A

PROGRAM DESCRIPTION AND FEATURES

The sequential Extended Kalman Filter routine used for torpedo tracking is modularized for ease of implementation. The program is general in nature and many of the parameters of the filter are variable including:

- a. The number of states in the filter--N
- b. The number of random forcing functions--M
- c. The number of measurements--JS
- d. Number of time slots--JTIME

The constant matrices PHI, R, COVW, and GAMMA are initialized in the beginning of the program using data statements. The filter is initialized with P(1/0) and X(1/0) (initial covariance of estimation error and states) using subroutine INIT. The first estimate is at time 1 and continues until ITIME = JTIME+1. True measurement times (ZI) are computed using either subroutines TRAJEC or TRAJC3, depending on whether single array or multiple array tracking is implemented. Either subroutine will compute four measurement times (T_C , T_X , T_Y , T_Z) for each time slot. The measurement times are corrupted by zero-mean, white Gaussian noise using the IBM-360 subroutine SNORM. For each of the four time measurements the corresponding row of the linearizing H matrix is calculated using either subroutine CHROW or CHROW3, depending on whether single array

or multiple array tracking is used. The corresponding gain matrix column GI is then found. These row and column values are utilized in forming the covariance of estimation error, PI, for that particular time measurement. Next the estimate of the observation time ZHAT from that particular hydrophone is formed using the subroutine CZHAT or CZHAT3, depending on whether single or multiple array tracking is implemented. The residual ZDIFF(I) = ZIC(I) - ZHAT, is then calculated. Finally the estimate of the states XI based on one time measurement is calculated and the process is repeated for the next measurement. After the four iterations, four transit time measures are formed using the last estimate of the states, X4, in either subroutine CZHAT or CZHAT3. The corresponding four transit time residuals are then calculated. The average of the absolute value of the time residuals is formed (ZDIFAV) and compared to a preset threshold value in the software. If ZDIFAV is above the threshold, subroutine QFIND is used to calculate a new Q which is then added to the last covariance of error estimate, P4. The software then reprocesses the transit time measurements for the same time slot. This procedure continues until the ZDIFAV measurement is equal or below the threshold value, at which time the last estimate of the state vector and covariance matrix become the XKK and PKK values for the time slot. Finally, the predictions XKKMI and PKKMI are formed before the process is repeated for the next time slot. PTLOP is used to generate line printer plots and PLOTG is used to generate VERSATEC plots.

A. PROGRAM SUBROUTINES

A brief description of the subroutines that are used with the filter are described below:

1. TRAJEC--This subroutine develops the torpedo trajectory which is used as truth data for the filter. The subroutine generates constant velocity and heading tracks. Turns of any desired turn rate and G force can be performed at any time along the torpedo track. The subroutine outputs true transit times, $ZI(I)$, and the X, Y, Z positions, $TD(I)$, of the torpedo for each time slot. The subroutine is used during single array tracking.
2. TRAJC3--This subroutine performs the same function as TRAJEC but is used only during multiple array tracking.
3. INIT--This subroutine generates the initial state vector ($X0/-1$) and initial covariance matrix ($P\ 0/-1$).
4. CHROW--This subroutine computes the appropriate row of the linearizing H matrix. Each row corresponds to one of the four transit time measurements, T_C, T_X, T_Y, T_Z . This subroutine is used during single state array tracking.
5. CHROW3--This subroutine performs the same functions as CHROW and is used during multiple array tracking.
6. CZHAT--This subroutine computes the estimated transit times for the filter. Four time estimates, ZHAT, are calculated corresponding to each of the four true transit times $ZI(I)$. This subroutine is used during single array tracking.
7. CZHAT3--Subroutine performs same functions as CZHAT however it is used only during multiple array tracking.

8. QFIND--This subroutine develops an adaptive Q matrix which is a function of the torpedo velocity. Three input variables defined in a data statement at the beginning of the program can be adjusted:

aa. SIGACC--Maximum expected horizontal acceleration of the torpedo.

bb. SIGDIV--Maximum expected change in vertical velocity.

cc. SIGCC--Maximum expected turn rate of the torpedo in the horizontal plane.

The values listed in the program were used and kept constant during the simulation tests. If the user desires not to use the adaptive Q subroutine software code is provided at the beginning of the program to calculate a constant Q matrix.

9. SNORM--This is an IBM-360 subroutine contained in the IMSL library. The routine generates zero mean white Gaussian noise with an RMS value normalized to 1. The main program scales the noise and adds it to the transit times measurements.

10. PLOTP--This is an IBM-360 subroutine used to generate the line printer plots. Information on this subroutine can be obtained from the IMSL library.

11. PLOTG--This is an IBM-360 subroutine used to generate the VERSATEC plots. Information on this subroutine can be obtained from the IMSL library.

B. UTILITY PROGRAMS

These subroutines were designed to be used for repetitive matrix and vector manipulations:

1. PROD--multiplying two matrices
2. MMULT--multiplying a matrix and a vector
3. VMULT--multiplying two vectors
4. TRANS--transposing a matrix
5. ADD--adding two matrices

APPENDIX B SEQUENTIAL EXTENDED KALMAN FILTER PROGRAM LISTING

```

DOUBLE PRECISION XKKM1,PKKM1,PKK,PKK,PHI,XKK,PHI,GAMMA,R,PHIPKK,PHIT,
1PKTEMP,HROW,PI,PDUM,QTEMP,GNUM,G1,XI,ZI,GAMMAT,Q,COVM,ZC,ZIC;
2ZCS,ZHAT,DATA,ZDIFF,COENOM,GDTEMP,ZDIFFAV
3COMMON XKKM1(5),PKKM1(5,5),PKK(5,5),PHIPKK(5,5),PHIT(5,5),
4DIMENSION PHI(5,5),GAMMA(5,3),R(4,4),PHIPKK(5,5),PHIT(5,5),
5PKTEMP(5,5),HROW(5,5),PI(5,5),PDUM(5,5),Q(5,5),COVM(3,3),
6GNUM(5),G1(5),XI(5),ZI(4),GAMMAT(3,5),P1(5,5),X3(5,10),
7X5(5,10),KOUNT(5,10),AKOUNT(5,10),PVEC(5,10),C1(2),C2(2),C3(2),
8TRUX(5,10),TRUY(5,10),ID(3),XDIFFI(5,10),XDIFFF3(5,10),ZDIFF(4),
9XC4(2),C5(2),C6(2),XP(5,5),XD(3),YB(4),ZB(4),DATA(23),ZIC(4)
10GX(5,10,4),GY(5,10,4),ZDIFF1(5,10,4),ZDM(2),GYM(2),ZIC(4)
11INITIALIZE CONSTANTS,HYDROPHONE MATRIX,PHI,R,COVM,GAMMA,
12AND DATA FOR TRAJECTORY
13DATA N/5,M/3,JS/4,IS/4,18/1,NZDIFF/4,JTIME/454/,
14SIGACC/36.2/,SIGDIV/1.0/,SIGCC/22.2/,ZC/4*0.0/,
15HYDRO/36000.,30000.,24000.,18000.,12000.,6000.,6*6000.,
166*0.0,36030.,30030.,24030.,18030.,12030.,6030.,6*6000.,
176*0.0,36000.,30000.,24000.,18000.,12000.,6000.,6*6030.,
186*0.0,36000.,30000.,24000.,18000.,12000.,7*6000.,6*30./
19DATA PHI/1.0,4*0.0,1.0E-10,4*0.0,1.0E-10,4*0.0,1.0E-10/,
20R/1.0E-10,4*0.0,1.0E-10,4*0.0,1.0E-10,4*0.0,1.0E-10/,
21COVM/1036.84,3*0.0,1036.84,3*0.0,1.0E-10,4*0.0,1.0E-10/,
22GAMMA/.86,1.31,5*0.0,.86,1.31,5*0.0,1.31/,
23
24DATA STATEMENT CONTAINS VALUES FOR TRAJEC OR TRAJC3
25DATA STATEMENT DEFINED IN SUBROUTINE TRAJEC OR TRAJC3
26
274DATA/39000.,6050.,300.,-60.,0.0,3*0.0,
2813*0.0,4.71,1.74,1.8,1.600.,800./
29SIGCC=SIGCC*3.14159/180.
30
31THE NEXT TWO STATEMENTS ARE REQUIRED FOR
32SUBROUTINE SNORM WHICH GENERATES WHITE NOISE
33
34CALL OVFLOW
35I9=6735476
36
37LOAD X(0/-1),P(0/-1)
38
39CALL INIT(XKKM1,PKKM1)
40
41GET TRANSPSES
42CALL TRANS(GAMMA,N,M,GAMMAT)
43CALL TRANS(PHI,N,N,PHIT)
44WRITE(6,163)

```

```

163      FORMAT('0','R MATRIX')
      DO 264 I=1,4
264      WRITE(6,136)(R(I,J),J=1,4)
136      FORMAT(4F14.11)
      WRITE(6,265)
      FORMAT('0','COV W MATRIX')
C265      DO 266 I=1,3
C266      WRITE(6,267)(COVW(I,J),J=1,3)
C267      FORMAT(3F11.6)
C      CALCULATE THE Q MATRIX
C      CALL PROD(GAMMA,COVW,N,M,M,QTEMP)
C      CALL PROD(QTEMP,GAMMAT,N,M,N,Q)
C      WRITE(6,100)
C100      FORMAT('0','Q MATRIX')
C      DO 101 LI=1,N
C101      WRITE(6,102)(Q(LI,LJ),LJ=1,N)
C102      FORMAT(1X,5F15.4)
C      START THE TIME SLOT LOOP AND SET ARRAY HANDOFF POINT
      ITIME=JTIME+1
      XT=36000.
      SW=XT-3000.
      DO 99 KK=1,ITIME
          WRITE(6,6) KK
          FORMAT('//1X,100('*,'),TIME=*,14//)
C      GET HYDROPHONE ARRAY COORDINATES
      DO 600 I3=1,4
          XB(I3)=HYDRO(I8,3*I3-2)
          YB(I3)=HYDRO(I8,3*I3-1)
          ZB(I3)=HYDRO(I8,3*I3)
          IF(XKKM1(I).GT.SW)GO TO 610
          I8=I8+1
          WRITE(6,759)I8,KK
          FORMAT('0',15('*,'),'ARRAY',1X,12,'STARTS TRACKING AT TIME',1X,13,
15('*,'))//)
          SW=SW-6000.
          XT=XT-6000.
          WRITE(6,67)
          FORMAT('0','XKKM1')
          DO 51 LA=1,N
          WRITE(6,50) XKKM1(LA)
          FORMAT(1X,5F14.4)
C67
C51
C50
C

```

```

THE00240
THE00245
THE00250
THE00255
THE00260
THE00265
THE00270
THE00275
THE00280
THE00285
THE00290
THE00295
THE00300
THE00305
THE00310
THE00315
THE00320
THE00325
THE00330
THE00335
THE00340
THE00345
THE00350
THE00355
THE00360
THE00365
THE00370
THE00375
THE00380
THE00385
THE00390
THE00395
THE00400
THE00405
THE00410
THE00415
THE00420
THE00425
THE00430
THE00435
THE00440
THE00445
THE00450
THE00455
THE00460
THE00465
THE00470
THE00475

```

```

C COMPUTE THE TRUE TIMES AND TRUE POSITIONS
C
C610 CALL TRAJC3(KK,DATA,ZI,TD,XB,YB,ZB)
C610 CALL TRAJEC(KK,DATA,ZI,TD)
      A14=PHI(1,2)
      TRUX(KK)=TD(1)
      TRUY(KK)=TD(2)
      TRUZ(KK)=TD(3)
C
C FIRST GET HROW-CALCULATE GAIN,ESTIMATE,COVARIANCE OF
C ERROR BASED ON ONE TIME MEASUREMENT-TC,TX,TY,TZ/
C
      DO 97 I=1,JS
      CALL CHROW3(I,HROW,XB,YB,ZB)
      CALL CHROW(I,HROW)
      CALL MMULT(PKKM1,HROW,N,N,GNUM)
      CALL VMULT(HROW,GNUM,N,GTEMP)
      GDENOM=GDTEMP+R(I,I)
      DO 16 IX=1,N
      GI(IX)=GNUM(IX)/GDENOM
      THIS IS THE FIRST GAIN COLUMN
      CALCULATE THE COVARIANCE OF ERROR PI
C
      DO 77 IP=1,N
      DO 79 JP=1,N
      PDUM(IP,JP)=(-1.*GI(IP))*HROW(JP)
      IF(IP.EQ.JP)PDUM(IP,JP)=1.+PDUM(IP,JP)
      CONTINUE
      CONTINUE
      CALL PROD (PDUM,PKKM1,N,N,N,PI)
      CALCULATE FIRST MEASUREMENT PREDICTION
C
      CALL CZHAT(I,ZHAT)
      CALL CZHAT3(I,ZHAT,XB,YB,ZB)
C
      GET WHITE NOISE,SCALE AND ADD TO TRUE MEASUREMENT TIME
C
      CALL SNORM(I9,ZC,4)
      ZCS=ZC(1)*.00001
      ZIC(I)=ZCS+ZI(I)
      ZDIFF(I)=ZIC(I)-ZHAT
C
      THESE STATEMENTS ARE FOR ADDITIONAL PLOTTING
C
      GX(KK,I)=GI(1)
      GY(KK,I)=GI(3)

```

```

THE00480
THE00485
THE00490
THE00495
THE00500
THE00505
THE00510
THE00515
THE00520
THE00525
THE00530
THE00535
THE00540
THE00545
THE00550
THE00555
THE00560
THE00565
THE00570
THE00575
THE00580
THE00585
THE00590
THE00595
THE00600
THE00605
THE00610
THE00615
THE00620
THE00625
THE00630
THE00635
THE00640
THE00645
THE00650
THE00655
THE00660
THE00665
THE00670
THE00675
THE00680
THE00685
THE00690
THE00695
THE00700
THE00705
THE00710
THE00715

```

THE00720
THE00725
THE00730
THE00735
THE00740
THE00745
THE00750
THE00755
THE00760
THE00765
THE00770
THE00775
THE00780
THE00785
THE00790
THE00795
THE00800
THE00805
THE00810
THE00815
THE00820
THE00825
THE00830
THE00835
THE00840
THE00845
THE00850
THE00855
THE00860
THE00865
THE00870
THE00875
THE00880
THE00885
THE00890
THE00895
THE00900
THE00905
THE00910
THE00915
THE00920
THE00925
THE00930
THE00935
THE00940
THE00945
THE00950
THE00955

```

ZDIFF1(KK,1)=ZDIFF(1)
IF ((KK<.GT.1).OR.(I.GT.1)) GO TO 73
ZDM(1)=ZDIFF(1)
ZDM(2)=ZDIFF(1)
GXM(1)=GI(1)
GXM(2)=GI(1)
GYM(1)=GI(3)
GYM(2)=GI(3)
GO TO 74
CONTINUE
IF (ZDIFF(1).GT.ZDM(1)) ZDM(1)=ZDIFF(1)
IF (ZDIFF(1).LT.ZDM(2)) ZDM(2)=ZDIFF(1)
IF (GI(1).GT.GXM(1)) GXM(1)=GI(1)
IF (GI(1).LT.GXM(2)) GXM(2)=GI(1)
IF (GI(3).GT.GYM(1)) GYM(1)=GI(3)
IF (GI(3).LT.GYM(2)) GYM(2)=GI(3)
CONTINUE
73
COMPUTE THE GATE FOR ERRONEOUS TIME MEASUREMENTS
PK1=DABS(PI(1,1))
PK3=DABS(PI(3,3))
PK5=DABS(PI(5,5))
IF ((PK1.GE.PK3).AND.(PK1.GE.PK5)) P=PK1
IF ((PK3.GE.PK1).AND.(PK3.GE.PK5)) P=PK3
IF ((PK5.GE.PK1).AND.(PK5.GE.PK3)) P=PK5
PGATE=SQRT(P)/4860.
RGATE=DSQRT(DABS(R(1,1)))
GATE=3.*(PGATE+RGATE)
EDIT INVALID TIME MEASUREMENTS
IF (ZDIFF(1).LT.GATE) GO TO 500
WRITE(6,501)KK
FORMAT(10,'GATE HAS BEEN EXCEEDED TIME',14)
DO 502 LG=1,N
GI(LG)=0.0
501
502
TAG INVALID TIME MEASUREMENT
ZDIFF(1)=999.
CALCULATE THE ESTIMATE BASED ON ONE MEASUREMENT
DO 17 IZ=1,N
XI(IZ)=XKKM1(IZ)+GI(IZ)*ZDIFF(1)
IF (I.EQ.4) GO TO 56
DO 19 IQ=1,N
500
17
12

```

THE00960
THE00965
THE00970
THE00975
THE00980
THE00985
THE00990
THE00995
THE01000
THE01005
THE01010
THE01015
THE01020
THE01025
THE01030
THE01035
THE01040
THE01045
THE01050
THE01055
THE01060
THE01065
THE01070
THE01075
THE01080
THE01085
THE01090
THE01095
THE01100
THE01105
THE01110
THE01115
THE01120
THE01125
THE01130
THE01135
THE01140
THE01145
THE01150
THE01155
THE01160
THE01165
THE01170
THE01175
THE01180
THE01185
THE01190
THE01195

```

19      XKKM1(IQ)=XI(IQ)
      DO 23 IQ=1,N
18      DO 18 JQ=1,N
23      PKKM1(IQ,JQ)=PI(IQ,JQ)
      CONTINUE
C31      WRITE(6,31)
97      FORMAT('O')
      CONTINUE

      NOTE-CALLED ORIGINAL X(O/-1),XKKM1. UPDATED AFTER 1
      MEASUREMENT CALLED IT XI, THEN MADE XKKM1=XI AND WENT
      THRU ITERATION AGAIN. AFTER YOU HAVE UPDATED XKKM1 FOR
      EACH MEASUREMENT XKK=XI AND PKK=PI

      DO 57 ID=1,N
      XKK(ID)=XI(ID)
      XKKM1(ID)=XI(ID)
      DO 58 JD=1,N
      PKK(ID,JD)=PI(ID,JD)
      PKKM1(ID,JD)=PI(ID,JD)
      CONTINUE

      RECALCULATE TIME MEASUREMENTS,FORM ABSOLUTE VALUE OF RESIDUALS

      DO 81 I=1,4
      EDIT INVALID TIME MEASUREMENTS FOR ADAPTIVE MANEUVER ROUTINE

      IF (ZDIFF(1).GE.999.) GO TO 82
      CALL CZHAT(1,ZHAT)
      CALL CZHAT3(1,ZHAT,XB,YB,ZB)
      ZDIFF(1)=DABS(ZIC(1)-ZHAT)
      GO TO 81
      ZDIFF(1)=0.0
      NZDIFF = NZDIFF - 1
      CONTINUE

      IF ALL TIME MEASUREMENTS EXCEED GATE BYPASS ADAPTIVE
      MANEUVER ROUTINE

      IF (NZDIFF.EQ.0) GO TO 80
      ZDIFAV=(ZDIFF(1)+ZDIFF(2)+ZDIFF(3)+ZDIFF(4))/NZDIFF
      DC NOT PERFORM ADAPTIVE MANEUVERING
      IF FILTER HAS NOT ACHIEVED STEADY STATE
      IF(KK.LE.4)GO TO 80

```



```

C      IF ZDIFAV MEETS CRITERIA TRANSFER OUT OF
C      ADAPTIVE MANEUVER ROUTINE
C      IF(ZDIFAV.LE..10000D-04)GO TO 80
C      INCREASE THE GAIN
C      CALL QFIND(KK,SIGACC,SIGDIV,SIGCC,A14,Q)
C      CALL ADD(PKK,Q,N,N,PKKM1)
C      PERFORM ADAPTIVE MANEUVERING BY REITERATING SAME TIME SLOT
C      GO TO 711
C
C      NZDIFF=4
C      XDIF1(KK)=XK(1)-TRUX(KK)
C      XDIF3(KK)=XK(3)-TRUY(KK)
C      XDIF5(KK)=XK(5)-TRUZ(KK)
C
C      CALCULATE THE PREDICTIONS FOR PKKM1
C      CALL QFIND(KK,SIGACC,SIGDIV,SIGCC,A14,Q)
C      WRITE(5,100)
C      DO 765 I=1,N
C      WRITE(6,766)(Q(I,J),J=1,N)
C      FORMAT(1X,5F10.4)
C      WRITE(6,767)
C      FORMAT(1X,1)
C
C      CALL PROD(P-I,PKK,N,N,N,PHIPKK)
C      CALL PROD(PHIPKK,PHIT,N,N,N,PKTEMP)
C      CALL ADD(PKTEMP,Q,N,N,PKKM1)
C
C      CALCULATE NEW XKKM1
C      CALL MMULT(PHI,XKK,N,N,XKKM1)
C      DO 41 IG=1,N
C      XP(IG,KK)=XKK(IG)
C      DO 38 II=1,N
C      DO 39 JJ=1,N
C      PI(KK,II,JJ)=PKK(II,JJ)
C      CONTINUE
C      CONTINUE
C      WRITE(6,909)ZDM(1),ZDM(2),GXM(1),GXM(2),GYM(2)
C      FORMAT(6E14.5)
C
C      SET UP ARRAYS AND SCALE FACTORS FOR PLOTP AND PLOTG

```

```

THE01200
THE01205
THE01210
THE01215
THE01220
THE01225
THE01230
THE01235
THE01240
THE01245
THE01250
THE01255
THE01260
THE01265
THE01270
THE01275
THE01280
THE01285
THE01290
THE01295
THE01300
THE01305
THE01310
THE01315
THE01320
THE01325
THE01330
THE01335
THE01340
THE01345
THE01350
THE01355
THE01360
THE01365
THE01370
THE01375
THE01380
THE01385
THE01390
THE01395
THE01400
THE01405
THE01410
THE01415
THE01420
THE01425
THE01430
THE01435

```



```

806 FORMAT(25X,'DEVIATION FROM TRUE Y')
   WRITE(6,900)
   CALL PLOTP(C3,C1,2,1)
   CALL PLOTP(AKOJNT,X5,ITIME,2)
   CALL PLOTP(AKOJNT,TRUZ,ITIME,3)
   WRITE(6,803)
803 FORMAT(25X,'Z DEPTH VS TIME')
   WRITE(6,900)
   CALL PLOTP(C4,C5,2,1)
   CALL PLOTP(X1,X3,ITIME,2)
   CALL PLOTP(TRUX,TRUY,ITIME,3)
   WRITE(6,804)
804 FORMAT(25X,'GEOGRAPHIC Y VS X')
   WRITE(6,900)
   THIS IS THE BLOCK TO DO THE EXTRA PLOTS
   CALL PLOTP(C3,ZDM,2,1)
   CALL PLOTP(AKOJNT,ZDIFFF1(1,1),ITIME,2)
   CALL PLOTP(AKOJNT,ZDIFFF1(1,2),ITIME,2)
   CALL PLOTP(AKOJNT,ZDIFFF1(1,3),ITIME,2)
   CALL PLOTP(AKOJNT,ZDIFFF1(1,4),ITIME,3)
   WRITE(6,70)
   WRITE(6,75)
   WRITE(6,900)
   CALL PLOTP(C3,GXM,2,1)
   CALL PLOTP(AKOJNT,GX(1,1),ITIME,2)
   CALL PLOTP(AKOJNT,GX(1,2),ITIME,2)
   CALL PLOTP(AKOJNT,GX(1,3),ITIME,2)
   CALL PLOTP(AKOJNT,GX(1,4),ITIME,3)
   WRITE(6,75)
   WRITE(6,900)
   CALL PLOTP(C3,GYM,2,1)
   CALL PLOTP(AKOJNT,GY(1,1),ITIME,2)
   CALL PLOTP(AKOJNT,GY(1,2),ITIME,2)
   CALL PLOTP(AKOJNT,GY(1,3),ITIME,2)
   CALL PLOTP(AKOJNT,GY(1,4),ITIME,3)
   WRITE(6,78)
   WRITE(6,75)
   WRITE(6,900)
   FORMAT(20X,'ZDIFF VS TIME ')
70 FORMAT(2X,'TC=+',TV=X,TZ=. ')
75 FORMAT(20X,'GAIN IN X VS TIME ')
76 FORMAT(20X,'GAIN IN Y VS TIME ')
78 END OF BLOCK FOR EXTRA PLOTS
C STOP
C C GENERATE PLOTG CURVES
C C

```

```

C
II=1
JJ=1
DO 240 KK=1, ITIME
PVEC(KK)=PI(KK, II, JJ)
240 PVEC(KK)=SQRT(PVEC(KK))
CALL PLOTG(AKOUNT, PVEC, ITIME, 1, 1, 1, 'TIME SLOT', 9, 'RMS ESTIMATE OF
1 POSITION ERROR(FT)', 34, 0, 0, 0, 15, 12, 7.)
II=3
JJ=3
DO 242 KK=1, ITIME
PVEC(KK)=PI(KK, II, JJ)
242 PVEC(KK)=SQRT(PVEC(KK))
CALL PLOTG(AKOUNT, PVEC, ITIME, 2, 1, 4, 'TIME SLOT', 9, 'RMS ESTIMATE OF
1 POSITION ERROR(FT)', 34, 0, 0, 0, 15, 12, 7.)
II=2
JJ=2
DO 241 KK=1, ITIME
PVEC(KK)=PI(KK, II, JJ)
241 PVEC(KK)=SQRT(PVEC(KK))
CALL PLOTG(AKOUNT, PVEC, ITIME, 1, 1, 1, 'TIME SLOT', 9, 'RMS ESTIMATE OF
1 VELOCITY ERROR(FT/SEC)', 38, 0, 0, 0, 0, 12, 7.)
II=4
JJ=4
DO 243 KK=1, ITIME
PVEC(KK)=PI(KK, II, JJ)
243 PVEC(KK)=SQRT(PVEC(KK))
CALL PLOTG(AKOUNT, PVEC, ITIME, 2, 1, 4, 'TIME SLOT', 9, 'RMS ESTIMATE OF
1 VELOCITY ERROR(FT/SEC)', 38, 0, 0, 0, 0, 12, 7.)
II=5
JJ=5
DO 244 KK=1, ITIME
PVEC(KK)=PI(KK, II, JJ)
244 PVEC(KK)=SQRT(PVEC(KK))
CALL PLOTG(AKOUNT, PVEC, ITIME, 1, 1, 1, 'TIME SLOT', 9, 'RMS ESTIMATE OF
1 DEPTH ERROR(FT)', 31, 0, 0, 0, 15, 12, 7.)
1 CALL PLOTG(AKOUNT, X1, ITIME, 1, 1, 1, 'TIME SLOT', 9, 'X-POSITION(FT)',
114, 0, 0, 0, 0, 12, 7.)
1 CALL PLOTG(AKOUNT, TRUX, ITIME, 2, 1, 4, 'TIME SLOT', 9, 'X-POSITION(FT)',
114, 0, 0, 0, 0, 12, 7.)
1 CALL PLOTG(AKOUNT, XDIFF1, ITIME, 1, 1, 1, 'TIME SLOT', 9, 'ERROR IN POSIT
1ION(FT)', 21, 0, 0, 0, -20, 12, 7.)
1 CALL PLOTG(AKOUNT, XDIFF3, ITIME, 2, 1, 4, 'TIME SLOT', 9, 'ERROR IN POSIT
1ION(FT)', 21, 0, 0, 0, -20, 12, 7.)
1 CALL PLOTG(AKOUNT, X3, ITIME, 1, 1, 1, 'TIME SLOT', 9, 'Y-POSITION(FT)', 14
1, 0, 0, 0, 0, 12, 7.)
1 CALL PLOTG(AKOUNT, TRUY, ITIME, 2, 1, 4, 'TIME SLOT', 9, 'Y-POSITION(FT)',
114, 0, 0, 0, 0, 12, 7.)
THE01920
THE01925
THE01930
THE01935
THE01940
THE01945
THE01950
THE01955
THE01960
THE01965
THE01970
THE01975
THE01980
THE01985
THE01990
THE01995
THE02000
THE02005
THE02010
THE02015
THE02020
THE02025
THE02030
THE02035
THE02040
THE02045
THE02050
THE02055
THE02060
THE02065
THE02070
THE02075
THE02080
THE02085
THE02090
THE02095
THE02100
THE02105
THE02110
THE02115
THE02120
THE02125
THE02130
THE02135
THE02140
THE02145
THE02150
THE02155

```

```

CALL PLOTG(AKOUNT,X5,ITIME,1,1,1,'TIME SLOT',9,'Z-DEPTH(FT)',11, THE02160
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,TRUZ,ITIME,2,1,4,'TIME SLOT',9,'Z-DEPTH(FT)',11, THE02165
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,XDIFF5,ITIME,1,1,1,'TIME SLOT',9,'ERROR IN Z-DEPT THE02170
1TH(FT)',20.0,0.0,12.7.)
CALL PLOTG(TRUX,ITIME,1,1,1,'X-POSITION(FT)',14,'Y-POSITION(FT) THE02175
11.0,0.0,12.7.)
CALL PLOTG(C3,ZDM,2,1,0,0,'TIME SLOT',9,'TIME RESIDUALS(SEC)',19, THE02180
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,ZDIFF1(1,1),ITIME,2,1,1,'TIME SLOT',9,'TIME RESI THE02185
DUALS(SEC)',19.0,0.0,12.7.)
CALL PLOTG(AKOUNT,ZDIFF1(1,2),ITIME,3,1,2,'TIME SLOT',9,'TIME RESI THE02190
DUALS(SEC)',19.0,0.0,12.7.)
CALL PLOTG(AKOUNT,ZDIFF1(1,3),ITIME,4,1,3,'TIME SLOT',9,'TIME RESI THE02195
DUALS(SEC)',19.0,0.0,12.7.)
CALL PLOTG(AKOUNT,ZDIFF1(1,4),ITIME,5,1,4,'TIME SLOT',9,'TIME RESI THE02200
DUALS(SEC)',19.0,0.0,12.7.)
CALL PLOTG(C3,GXM,2,1,0,0,'TIME SLOT',9,'X-GAIN',6,0,0.0,0.0., THE02205
112.7.)
CALL PLOTG(AKOUNT,GX(1,1),ITIME,2,1,1,'TIME SLOT',9,'X-GAIN',6,0., THE02210
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GX(1,2),ITIME,3,1,2,'TIME SLOT',9,'X-GAIN',6,0., THE02215
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GX(1,3),ITIME,4,1,3,'TIME SLOT',9,'X-GAIN',6,0., THE02220
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GX(1,4),ITIME,5,1,4,'TIME SLOT',9,'X-GAIN',6,0., THE02225
10.0,0.0,12.7.)
CALL PLOTG(C3,GYM,2,1,0,0,'TIME SLOT',9,'Y-GAIN',6,0,0.0,0.0., THE02230
112.7.)
CALL PLOTG(AKOUNT,GY(1,1),ITIME,2,1,1,'TIME SLOT',9,'Y-GAIN',6,0., THE02235
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,2),ITIME,3,1,2,'TIME SLOT',9,'Y-GAIN',6,0., THE02240
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,3),ITIME,4,1,3,'TIME SLOT',9,'Y-GAIN',6,0., THE02245
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,4),ITIME,5,1,4,'TIME SLOT',9,'Y-GAIN',6,0., THE02250
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,1),ITIME,2,1,1,'TIME SLOT',9,'Y-GAIN',6,0., THE02255
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,2),ITIME,3,1,2,'TIME SLOT',9,'Y-GAIN',6,0., THE02260
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,3),ITIME,4,1,3,'TIME SLOT',9,'Y-GAIN',6,0., THE02265
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,4),ITIME,5,1,4,'TIME SLOT',9,'Y-GAIN',6,0., THE02270
10.0,0.0,12.7.)
CALL PLOTG(C3,GYM,2,1,0,0,'TIME SLOT',9,'Y-GAIN',6,0,0.0,0.0., THE02275
112.7.)
CALL PLOTG(AKOUNT,GY(1,1),ITIME,2,1,1,'TIME SLOT',9,'Y-GAIN',6,0., THE02280
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,2),ITIME,3,1,2,'TIME SLOT',9,'Y-GAIN',6,0., THE02285
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,3),ITIME,4,1,3,'TIME SLOT',9,'Y-GAIN',6,0., THE02290
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,4),ITIME,5,1,4,'TIME SLOT',9,'Y-GAIN',6,0., THE02295
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,1),ITIME,2,1,1,'TIME SLOT',9,'Y-GAIN',6,0., THE02300
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,2),ITIME,3,1,2,'TIME SLOT',9,'Y-GAIN',6,0., THE02305
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,3),ITIME,4,1,3,'TIME SLOT',9,'Y-GAIN',6,0., THE02310
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,4),ITIME,5,1,4,'TIME SLOT',9,'Y-GAIN',6,0., THE02315
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,1),ITIME,2,1,1,'TIME SLOT',9,'Y-GAIN',6,0., THE02320
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,2),ITIME,3,1,2,'TIME SLOT',9,'Y-GAIN',6,0., THE02325
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,3),ITIME,4,1,3,'TIME SLOT',9,'Y-GAIN',6,0., THE02330
10.0,0.0,12.7.)
CALL PLOTG(AKOUNT,GY(1,4),ITIME,5,1,4,'TIME SLOT',9,'Y-GAIN',6,0., THE02335
10.0,0.0,12.7.)
CALL PLOT(0.0,0.0,999)
STOP
END

```



```

C C C C C C C C C C
SUBROUTINE TRAJC3(K, DATA, ZI, TD, XB, YB, ZB)
SUBROUTINE COMPUTES TRUE TRAJECTORY OF TORPEDO
DATA(1)=TRUE X POSITION, DATA(2)=TRUE Y POSITION
DATA(3)=TRUE Z POSITION, DATA(4)=TRUE X VELOCITY
DATA(5)=TRUE Y VELOCITY, DATA(6)=0., DATA(7)=0., DATA(8)=0.
DATA(9)=0., DATA(10)=0., DATA(11)=0.
DATA(12)=ACCELERATION ANGLE(RADIANS), DATA(13)=TURN RATE(RAD./SEC)
DATA(14)=SAMPLE TIME, DATA(15)=HORIZONTAL ACCELERATION(FY/SEC)
DATA(16)=DATA(17)=TIME SLOT INTERVAL DJRING WHICH MANEUVER
TAKES PLACE, ZI=TRUE TIMES, TD=TRUE POSITION
DOUBLE PRECISION DATA ZI
DIMENSION DATA(23), ZI(4), TD(3), XB(4), YB(4), ZB(4)
T=0.0
VEL=4860.
RANGE=DSQRT(DATA(1)*DATA(1)+DATA(2)*DATA(2)+DATA(3)*DATA(3))
DO 5 I=1,4
  ZI(I)=1./VEL*((DATA(1))-XB(I))*2)+((DATA(2))-YB(I))*2)
  +((DATA(3))-ZB(I))*2))*0.5
1 CONTINUE
DO 9 I=1,3
  TD(I)=DATA(I)
CONTINUE
IF((K.LE.DATA(17)).AND.(K.GT.DATA(16))) GO TO 11
DATA(7)=0.0
DATA(8)=0.0
DATA(14)=1.31
GO TO 12
DATA(14)=.005
DATA(12)=DATA(12)+DATA(13)*DATA(14)
DATA(7)=DATA(15)*DCOS(DATA(12))
DATA(8)=DATA(15)*DSIN(DATA(12))
DO 10 I=1,5
  DATA(1)=DATA(1)+DATA(I+3)*DATA(14)+(((DATA(14))*2)/2)*DATA(I+6)
CONTINUE
T=T+DATA(14)
IF(ABS(T-1.31).LE.0.0001) RETURN
GO TO 13
END

```

```

THE02605
THE02610
THE02615
THE02620
THE02625
THE02630
THE02635
THE02640
THE02645
THE02650
THE02655
THE02660
THE02665
THE02670
THE02675
THE02680
THE02685
THE02690
THE02695
THE02700
THE02705
THE02710
THE02715
THE02720
THE02725
THE02730
THE02735
THE02740
THE02745
THE02750
THE02755
THE02760
THE02765
THE02770
THE02775
THE02780
THE02785
THE02790
THE02795
THE02800
THE02805

```

```

C THIS ROUTINE IS TO INITIALIZE THE ARRAYS XKKM1 AND PKKM1.
C THESE VARIABLES ARE PART OF A COMMON BLOCK.
C
SUBROUTINE INIT (XKKM1,PKKM1)
  DOUBLE PRECISION XKKM1,PKKM1
  DIMENSION XKKM1(5),PKKM1(5,5)
  DO 20 J=1,5
    DO 10 I=1,5
      PKKM1(I,J)=0.0
10  CONTINUE
      XKKM1(1)=38975.
      XKKM1(2)=-50.
      XKKM1(3)=6025.
      XKKM1(4)=-10.
      XKKM1(5)=300.
20  CONTINUE
      DO 30 I=1,5
        PKKM1(I,1)=1000.
30  CONTINUE
      RETURN
      END
THE02810
THE02815
THE02820
THE02825
THE02830
THE02835
THE02840
THE02845
THE02850
THE02855
THE02860
THE02865
THE02870
THE02875
THE02880
THE02885
THE02890
THE02895
THE02900
THE02905
THE02910
THE02915

```


THE02920
THE02925
THE02930
THE02935
THE02940
THE02945
THE02950
THE02955
THE02960
THE02965
THE02970
THE02975
THE02980
THE02985
THE02990
THE02995
THE03000
THE03005

SUBROUTINE CZHAT(I,ZHAT)
THIS SUBROUTINE COMPUTES ESTIMATES OF TRANSIT TIME MEASUREMENTS
FOR SINGLE ARRAY TRACKING
DOUBLE PRECISION XKKM1,ZHAT
COMMON XKKM1(5)
VEL=4860.
IF(I.EQ.1) ZHAT=1./VEL*((XKKM1(1)+15.)*2)+((XKKM1(3)+15.)*2)
1+((XKKM1(5)+15.)*2)**0.5
IF(I.EQ.2) ZHAT=1./VEL*((XKKM1(1)-15.)*2)+((XKKM1(3)+15.)*2)
1+((XKKM1(5)+15.)*2)**0.5
IF(I.EQ.3) ZHAT=1./VEL*((XKKM1(1)+15.)*2)+((XKKM1(3)-15.)*2)
1+((XKKM1(5)+15.)*2)**0.5
IF(I.EQ.4) ZHAT=1./VEL*((XKKM1(1)+15.)*2)+((XKKM1(3)+15.)*2)
1+((XKKM1(5)-15.)*2)**0.5
RETURN
END

C
C
C

THE03010
THE03015
THE03020
THE03025
THE03030
THE03035
THE03040
THE03045
THE03050
THE03055
THE03060
THE03065
THE03070
THE03075
THE03080
THE03085

SUBROUTINE CZHAT3 (I,ZHAT,XB,YB,ZB)
THIS SUBROUTINE COMPUTES ESTIMATES OF TRANSIT TIME MEASUREMENTS
FOR MULTIPLE ARRAY TRACKING
DOUBLE PRECISION XKKM1,ZHAT
COMMON XKKM1(5)
DIMENSION XB(4),YB(4),ZB(4)
VEL=4860.
XO=XB(1)
YO=YB(1)
ZO=ZB(1)
ZHAT=((XKKM1(1)-XO)**2)+((XKKM1(3)-YO)**2)+((XKKM1(5)
1-ZO)**2)**0.5
RETURN
END

C
C
C
C

THE03090
THE03095
THE03100
THE03105
THE03110
THE03115
THE03120
THE03125
THE03130
THE03135
THE03140
THE03145
THE03150
THE03155
THE03160
THE03165
THE03170
THE03175
THE03180
THE03185
THE03190
THE03195
THE03200
THE03205
THE03210
THE03215
THE03220
THE03225
THE03230
THE03235
THE03240
THE03245
THE03250
THE03255
THE03260
THE03265
THE03270

```

SUBROUTINE CHROW(I,HROW)
  THIS SUBROUTINE USED FOR SINGLE ARRAY TRACKING
  DOUBLE PRECISION XKKM1,DENOM1,DENOM2,DENOM3,DENOM4,DENOM,HROW,
  H1,H3,H5
  COMMON XKKM1(5)
  DIMENSION HROW(5)
  VEL=4860.
  DENOM1=((XKKM1(1)+15.)*2)+((XKKM1(3)+15.)*2)+((XKKM1
  1(5)+15.)*2)**0.5
  DENOM2=((XKKM1(1)-15.)*2)+((XKKM1(3)+15.)*2)+((XKKM1
  1(5)+15.)*2)**0.5
  DENOM3=((XKKM1(1)+15.)*2)+((XKKM1(3)-15.)*2)+((XKKM1
  1(5)+15.)*2)**0.5
  DENOM4=((XKKM1(1)+15.)*2)+((XKKM1(3)+15.)*2)+((XKKM1
  1(5)-15.)*2)**0.5
  A1=1.
  A2=1.
  A3=1.
  DENOM=DENOM1
  IF(1.EQ.2)DENOM=DENOM2
  IF(1.EQ.3)DENOM=DENOM3
  IF(1.EQ.4)DENOM=DENOM4
  IF(1.EQ.2)A1=-1.
  H1=(1./VEL)*((XKKM1(1)+A1*15.)/DENOM)
  IF(1.EQ.3)A2=-1.
  H3=(1./VEL)*((XKKM1(3)+A2*15.)/DENOM)
  IF(1.EQ.4)A3=-1.
  H5=(1./VEL)*((XKKM1(5)+A3*15.)/DENOM)
  HROW(1)=H1
  HROW(3)=H3
  HROW(5)=H5
  DO 27 J=2,4,2
  HROW(J)=0.
  RETURN
  END

```

C
C
C

27

THE03275
THE03280
THE03285
THE03290
THE03295
THE03300
THE03305
THE03310
THE03315
THE03320
THE03325
THE03330
THE03335
THE03340
THE03345
THE03350
THE03355
THE03360
THE03365
THE03370

```

SUBROUTINE CHROW3 (I,HROW,XB,YB,ZB)
THIS SUBROUTINE USED FOR MULTIPLE ARRAY TRACKING

DOUBLE PRECISION XKKM1,HROW,DENOM
COMMON XKKM1(5)
DIMENSION HROW(5),XB(4),YB(4),ZB(4)
VEL=4860.
X0=XB(1)
Y0=YB(1)
Z0=ZB(1)
DENOM=((XKKM1(1)-X0)**2)+((XKKM1(3)-Y0)**2)+((XKKM1(5)-Z0)
1**2)**0.5
HROW(1)=(1./VEL)*((XKKM1(1)-X0)/DENOM)
HROW(3)=(1./VEL)*((XKKM1(3)-Y0)/DENOM)
HROW(5)=(1./VEL)*((XKKM1(5)-Z0)/DENOM)
DO 27 J=2,4,2
HROW(J)=0.
RETURN
END

```

C
C
C

27

```

C
C
SUBROUTINE QFIND(K,SIGACC,SIGDIV,SIGCC,A,Q)
THIS SUBROUTINE COMPUTES THE ADAPTIVE Q MATRIX
DOUBLE PRECISION XHKKM1,PKKM1,PKK,XHKK,Q
COMMON XHKKM1(5),PKKM1(5,5),PKK(5,5),XHKK(5)
DIMENSION Q(5,5)
IF(K.NE.1) GO TO 15
DO 10 I=1,5
DO 10 J=1,5
Q(I,J)=0.0
SIGACC=SIGACC**2
Q(5,5)=(SIGDIV*A)**2
SIGCC=SIGCC**2
G1=(A**2)/2.0
G2=G1**2
G3=A*G1
A2=A**2
10  A1=XHKK(2)**2+XHKK(4)**2
A3=XHKK(2)/SQRT(A1)
B=XHKK(4)
C=XHKK(4)/SQRT(A1)
D=XHKK(2)
E1=(A3**2)*SIGACC+(B**2)*SIGCC
E12=A3*C*SIGACC-B*D*SIGCC
E2=(C**2)*SIGACC+(D**2)*SIGCC
Q(1,1)=E1*G2
Q(1,2)=E1*G2
Q(1,3)=E12*G2
Q(1,4)=G3*E12
Q(2,2)=A2*E12
Q(2,3)=G3*E12
Q(2,4)=A2*E12
Q(3,3)=G2*E2
Q(3,4)=G3*E2
Q(4,4)=A2*E2
DO 27 I=1,4
DO 27 J=1,I
Q(I,J)=Q(J,I)
27  RETURN
END
THE03375
THE03380
THE03385
THE03390
THE03395
THE03400
THE03405
THE03410
THE03415
THE03420
THE03425
THE03430
THE03435
THE03440
THE03445
THE03450
THE03455
THE03460
THE03465
THE03470
THE03475
THE03480
THE03485
THE03490
THE03495
THE03500
THE03505
THE03510
THE03515
THE03520
THE03525
THE03530
THE03535
THE03540
THE03545
THE03550
THE03555
THE03560
THE03565
THE03570
THE03575

```

THE03580
 THE03585
 THE03590
 THE03595
 THE03600
 THE03605
 THE03610
 THE03615
 THE03620
 THE03625
 THE03630
 THE03635

SUBROUTINE PROD(A,B,N,M,L,C)
 DOUBLE PRECISION A,B,C
 DIMENSION A(N,M),B(M,L),C(N,L)
 DO 1 I=1,N
 DO 1 J=1,L
 C(I,J)=0.
 DO 2 I=1,N
 DO 2 J=1,L
 DO 2 K=1,M
 C(I,J)=C(I,J)+A(I,K)*B(K,J)
 RETURN
 END

1

2

```

SUBROUTINE MMJLT(A,B,N,M,C)
DOUBLE PRECISION A,B,C
DIMENSION A(N,M),B(M),C(N)
DO 3 I=1,N
  C(I)=0.
DO 4 J=1,M
  C(I)=C(I)+A(I,J)*B(J)
CONTINUE
RETURN
END

```

4 3

```

THE03640
THE03645
THE03650
THE03655
THE03660
THE03665
THE03670
THE03675
THE03680
THE03685

```

```

SUBROUTINE VMJLT(A,B,N,C)
DOUBLE PRECISION A,B
DIMENSION A(N),B(N)
C=0.
DO 6, I=1,N
C=C+A(I)*B(I)
RETURN
END

```

6

```

THE03690
THE03695
THE03700
THE03705
THE03710
THE03715
THE03720
THE03725

```



```

SUBROUTINE TRANS(A,N,M,B)
DOUBLE PRECISION A,B
DIMENSION A(N,M),B(M,N)
DO 13 I=1,N
DO 13 J=1,M
B(J,I)=A(I,J)
RETURN
END

```

13

```

THE03730
THE03735
THE03740
THE03745
THE03750
THE03755
THE03760
THE03765

```

THE03770
THE03775
THE03780
THE03785
THE03790
THE03795
THE03800
THE03805

SUBROUTINE ADD (A,B,N,M,C)
DOUBLE PRECISION A,B,C
DIMENSION A(N,M),B(N,M),C(N,M)
DO 15 I=1,N
DO 15 J=1,M
C(I,J)=A(I,J)+B(I,J)
RETURN
END

15

LIST OF REFERENCES

1. Benson, Eric J. An Application of Kalman Filtering to Underwater Tracking, Master's Thesis, Naval Postgraduate School, December 1976.
2. Dwyer, Dennis M. Real Time Kalman Filtering For Torpedo Range Tracking, Master's Thesis, Naval Postgraduate School, December 1978.
3. Technical Manual, NAVORD OD 41964, NAVTOPRSTA Keyport Range Complex and Associated Data, May 1970.
4. Kirk, Donald E., EE4413 Optimal Control Systems Class Notes, Unpublished.
5. Mitschang, George W. An Application of Nonlinear Filtering Theory to Passive Target Location and Tracking. Ph.D. Thesis, Naval Postgraduate School, 1975.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4. Professor H. A. Titus, Code 62Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	5
5. Associate Professor Alex Gerba, Code 62Gz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6. Commanding Officer Naval Torpedo Station Keyport, Washington 98345	5
7. Paul A. O'Brien ASATD-AT-51 Naval Air Test Center Patuxent River, Maryland 20670	2

DATE
FILMED
— 8